Automation and Remote Control, Vol. 65, No. 6, 2004, pp. 871–892. Translated from Avtomatika i Telemekhanika, No. 6, 2004, pp. 37–60.

ARITHMETIC LOGIC

Large Systems of Boolean Functions: Realization by Modular Arithmetic Methods

O. A. Fin'ko

Krasnodar, Russia

Received December 17, 2003

Abstract—Modular polynomial and spectral arithmetic-logical forms are introduced for representing Boolean functions so as to determine certain useful properties associated with the restricted number range (solution of the problem of large coefficients of arithmetical polynomials) in parallel logical computations.

1. INTRODUCTION

A need arises for processing large volumes of logical data in designing and analyzing discrete devices and logical controls for complicated technical systems and real time processes. Traditional methods of describing logical functions by Boolean formulas and Zhegalkin polynomials are ineffective for realizing such systems by the existing computing machines [1]. This inconsistency can be avoided through the methods of realization of Boolean functions by arithmetical polynomials, which yield a relation between logical and arithmetical data [1–3]. The classes of arithmetic-logical forms for Boolean functions based on the representation of Boolean functions in the spectral range are widened in [4–7]. This extension gives a relation between different forms of representation of Boolean functions and their design methods, and paves the way for application of effective mathematical and digital signal processing tools in analysis and synthesis of Boolean functions.

Certain advantages associated with the restricted number range of representation of intermediate results and parallelization of computation are helpful in deriving modular transformations [8–10]. In this paper, we study the construction of modular arithmetic forms of representation for Boolean functions. Its specific properties are helpful in extending the advantages of modular arithmetic to logical computations. In Section 2, we describe the properties of representation of Boolean functions by arithmetical polynomials and their derivation methods. These methods are used to construct modular arithmetic-logical forms in Sections 3 and 4. In Section 3, we introduce modular arithmetic-logical forms based on unimodular arithmetic and describe modular arithmetic-logical forms based on multimodular arithmetic (the Chinese remainder theorem) in Section. 4.

2. REPRESENTATION OF A SYSTEM OF BOOLEAN FUNCTIONS BY ARITHMETICAL POLYNOMIALS. FORMULATION OF THE PROBLEM

2.1. Theorem on the Representation of a System of Boolean Functions by One Arithmetical Polynomial

Let a d-output Boolean function $f(X)$ (a system of Boolean functions $f_1(X)$, $f_2(X)$, …, $f_d(X)$) of n variables $X = x_1, x_2, \ldots, x_n$

$$\begin{cases} y_1 = f_1(X) \\ \vdots \\ y_d = f_d(X) \end{cases} \tag{1}$$

0005-1179/04/6506-0871

Table 1

x_2	x_1	y_2	y_1	Y (decimal expression)
0	0	1	1	3
0	1	0	0	0
1	0	0	0	0
1	1	0	1	1

be given, where y_j is the value of the jth Boolean function $f_j(X)$ and $x_i, y_j \in \{0,1\}$ $(i = 1,\ldots,n,$ $j = 1,\ldots,d)$. The cortege of values $y_d * y_{d-1} * \ldots * y_1$ of the Boolean function, where the asterisk is the separator, is interpreted as the code of a nonnegative integer Y in binary representation:

$$y_d * y_{d-1} * \ldots * y_1 = Y = \sum_{j=1}^{d} y_j 2^{j-1}.$$

Example 1. The representation of Y corresponding to the system of Boolean functions

$$\begin{cases} f_1(X) = \overline{x_1 \oplus x_2} \\ f_2(X) = \overline{x_1 \vee x_2}, \end{cases} \tag{2}$$

is given in Table 1 (here and in what follows, the symbols $\vee$, $\wedge$, $\oplus$, and $\neg$ denote the operations of logical addition, multiplication, addition modulo 2, and inversion, respectively).

Theorem 1 ([1–3]). *A cortege of Boolean functions $f_d(X)*f_{d-1}(X)*\ldots*f_1(X)$ can be represented uniquely by an arithmetical polynomial*

$$Y = D(X) = \sum_{i=0}^{2^n-1} c_i x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \tag{3}$$

where $i_1 i_2 \ldots i_n = \sum\limits_{u=1}^{n} i_u 2^{n-u}$, $i_u \in \{0,1\}$; $x_u^{i_u} = \begin{cases} x_u, & i_u = 1 \\ 1, & i_u = 0 \end{cases}$, and $c_i \in Z$ $(i = 0,1,\ldots,2^n-1)$.

2.2. Algebraic Method of Derivation of Arithmetical Polynomials. Linear Arithmetical Polynomials

The algebraic method of derivation of the arithmetical polynomial (3) consists in realizing

Algorithm 1.

Step 1. Derivation of an arithmetical polynomial $P_j(X)$ for every Boolean function $y_j = f_j(X)$:

$$f_j(X) = P_j(X) = \sum_{i=0}^{2^n-1} r_{j,i} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \quad j = 1,\ldots,d. \tag{4}$$

Step 2. Derivation of weighted arithmetical polynomials of weights 2^{j-1} $(j = 1,\ldots,d)$:

$$P'_j(X) = P_j(X)2^{j-1} = \sum_{i=0}^{2^n-1} r'_{j,i} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \quad j = 1,\ldots,d, \tag{5}$$

where $r'_{j,i} = r_{j,i}2^{j-1}$ $(j = 1,\ldots,d;\ i = 0,1,\ldots,2^n-1)$.

Step 3. Derivation of the unknown arithmetical polynomial $D(X)$ (3) by summation of the coefficients of the arithmetical polynomial $P'_j(X)$ for all $j = 1, \ldots, d$:

$$D(X) = \sum_{i=0}^{2^n-1} \sum_{j=1}^{d} r'_{j,i} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} = \sum_{i=0}^{2^n-1} c_i x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}, \tag{6}$$

where $c_i = \sum\limits_{j=1}^{d} r'_{j,i}$ $(i = 0, 1, \ldots, 2^n - 1)$.

Example 2. For the system of Boolean functions (2), algorithm 1 is realized in three steps:

Step 1. Using the relations

$$x_1 \wedge x_2 = x_1 x_2,$$
$$x_1 \vee x_2 = x_1 + x_2 - x_1 x_2,$$
$$x_1 \oplus x_2 = x_1 + x_2 - 2x_1 x_2,$$
$$\overline{x} = 1 - x,$$

we obtain

$$f_1(X) = P_1(X) = \overline{x_1 \oplus x_2} = 1 - x_1 - x_2 + 2x_1 x_2,$$
$$f_2(X) = P_2(X) = \overline{x_1 \vee x_2} = 1 - x_1 - x_2 + x_1 x_2.$$

Step 2.

$$P'_1(X) = 2^0(1 - x_1 - x_2 + 2x_1 x_2) = 1 - x_1 - x_2 + 2x_1 x_2,$$
$$P'_2(X) = 2^1(1 - x_1 - x_2 + x_1 x_2) = 2 - 2x_1 - 2x_2 + 2x_1 x_2.$$

Step 3.

$$D(X) = 1 + 2 - (1 + 2)x_1 - (1 + 2)x_2 + (2 + 2)x_1 x_2 = 3 - 3x_1 - 3x_2 + 4x_1 x_2.$$

This example shows that the number range required for representing coefficients and results of intermediate computations of arithmetical polynomials may be far wider than the number range that is sufficient for representing Y. In our case, two binary digits $(0 \le Y \le 2^2 - 1)$ are sufficient for the representation of Y, whereas four binary digits $(-3 \le c_{1\ldots4} \le 5)$ are required for the representation of the coefficients $c_1, \ldots, c_4$ of the arithmetical polynomial, and the results of intermediate computations for $x_1 = x_2 = 1$ may take values in the range from -6 to $+7$.

Of great significance for the representation of d-output Boolean functions $f(X)$ are linear arithmetical polynomials $L(X)$, which are defined by the expression

$$U = L(X) = d_0 + \sum_{i=1}^{n} d_i x_i = d_0 + d_1 x_1 + \ldots + d_n x_n, \tag{7}$$

where the coefficients $d_0, d_1, \ldots, d_n$ are integers [1, 11, 12].

The functions $f_j(X)$ are computed with the masking operator $\Xi^t\{U\}$ [11, 12] used in determining the tth binary digit (output) of the representation $U = a_r 2^{r-1} + \ldots + a_t 2^{t-1} + \cdots + a_2 2^1 + a_1 2^0$, i.e., $\Xi^t\{U\} = a_t$.

Example 3. The arithmetical polynomial $P_j(X) = x_1 + x_2 - x_1x_2$ for the Boolean function $f_j(X) = x_1 \vee x_2$ is linearized by introducing an additional (garbage) Boolean function $f_j^{(1)}(X)$. A system of Boolean functions is formed

$$f_j^{(1)}(X) = 1 \oplus x_1 \oplus x_2,$$
$$f_j^{(2)}(X) = x_1 \vee x_2.$$

Hence $U = L(X) = 2^1 f_j^{(2)}(X) + 2^0 f_j^{(1)}(X) = 1 + x_1 + x_2$ and $f_j(X) = \Xi^2\{U\}$ [11].

Thus, the representation of systems of Boolean functions (1) by linear arithmetical polynomials $L(X)$ is based on the same weighting principle of representation of Boolean functions with weights 2^i $(i = 0, 1, \dots)$ as the construction of arithmetical polynomials $D(X)$ (3). But the value of i is chosen with regard for garbage Boolean functions.

Example 4 ([11]). A system of Boolean functions

$$\begin{cases} f_A(X) = x_1 \wedge x_3 \\ f_B(X) = \overline{x}_1 \wedge x_2 \\ f_C(X) = \overline{x}_2 \wedge x_3 \end{cases} \tag{8}$$

is given. For the resultant arithmetical polynomial to be linear, we must add garbage Boolean functions $f_A^{(1)}(X)$, $f_B^{(3)}(X)$, and $f_C^{(5)}(X)$ and thus obtain the system of Boolean functions

$$\begin{cases} f_A^{(1)}(X) = x_1 \oplus x_3 \\ f_A^{(2)}(X) = f_A(X), \end{cases} \quad \begin{cases} f_B^{(3)}(X) = \overline{x}_1 \oplus x_2 \\ f_B^{(4)}(X) = f_B(X), \end{cases} \quad \begin{cases} f_C^{(5)}(X) = \overline{x}_2 \oplus x_3 \\ f_C^{(6)}(X) = f_C(X). \end{cases}$$

Now, according to (7) and Example 3,

$$U_A = L'_A(X) = 2^1 f_A^{(1)}(X) + 2^0 f_A^{(2)}(X) = x_1 + x_3,$$
$$U_B = L'_B(X) = 2^1 f_B^{(3)}(X) + 2^0 f_B^{(4)}(X) = 1 - x_1 + x_2,$$
$$U_C = L'_C(X) = 2^1 f_C^{(5)}(X) + 2^0 f_C^{(6)}(X) = 1 - x_2 + x_3.$$

We obtain the linear arithmetical polynomial

$$U = L(X) = 2^0 L''_A(X) + 2^2 L'_B(X) + 2^4 L'_C(X) = 20 - 4x_1 - 12x_2 + 16x_3. \tag{9}$$

To determine the tth Boolean function, we apply the masking operator $\Xi^t\{Y\}$

$$f_A(X) = \Xi^2\{U\},$$
$$f_B(X) = \Xi^4\{U\},$$
$$f_C(X) = \Xi^6\{U\}.$$

The liner form of the arithmetical polynomial (9) is obtained by introducing garbage Boolean functions and increasing 2^3 times the number range required for the representation of U.

2.3. Matrix Transformations

By direct and inverse matrix transformations (Fourier discrete logical transformations), we mean the transformations [1, 6, 12]

$$\mathbf{C} = \mathbf{A}_{2^n}\mathbf{Y}, \tag{10}$$
$$\mathbf{Y} = \mathbf{A}^{-1}{}_{2^n}\mathbf{C}, \tag{11}$$

where $\mathbf{A}_{2^n}$ and $\mathbf{A}_{2^n}^{-1}$ are $2^n \times 2^n$ matrices of direct and inverse arithmetical transformations (transformation base), respectively, $\mathbf{Y}$ is the truth vector of the d-output Boolean function $f(X)$ for which $\mathbf{Y} = [\mathbf{Y}_d \,|\, \mathbf{Y}_{d-1} \,|\, \ldots \,|\, \mathbf{Y}_1]^{\mathrm{T}} = \left[Y^{(0)}Y^{(1)} \ldots Y^{(2^n-1)}\right]^{\mathrm{T}}$, T is the transpose, $Y^{(i)}$ is the numerical value of the dth output Boolean function $f(X)$ for the ith pattern of Boolean variables of the usual truth table (see Example 1), and $\mathbf{C} = [c_0\ c_1\ \ldots\ c_{2^n-1}]^{\mathrm{T}}$ is a vector of coefficients of the arithmetical polynomial (3) or the arithmetical spectrum of the Boolean function.

The matrix $\mathbf{A}_{2^n} = \left[\begin{array}{c|c} \mathbf{A}_{2^{n-1}} & 0 \\ \hline -\mathbf{A}_{2^{n-1}} & \mathbf{A}_{2^{n-1}} \end{array}\right]$ is the nth Kronecker degree $\mathbf{A}_{2^n} = \bigotimes_{j=1}^{n} \mathbf{A}_1$ of the base matrix $\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$ and $\mathbf{A}_{2^n}^{-1} = \bigotimes_{j=1}^{n} \mathbf{A}_1^{-1}$, where $\mathbf{A}_1^{-1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ is the base matrix of the inverse transformation. The matrix $-\mathbf{A}_{2^{n-1}}$ is formed from $\mathbf{A}_{2^{n-1}}$ by reversing the signs of unit elements.

Matrix transformations easily yield to algorithmization and are convenient in applications.

Example 5. Let there be given a three-output Boolean function. Vectors of its values are

$$\mathbf{Y}_1 = [01011011]^{\mathrm{T}},$$
$$\mathbf{Y}_2 = [01100111]^{\mathrm{T}},$$
$$\mathbf{Y}_3 = [01101001]^{\mathrm{T}}.$$

Therefore,

$$\mathbf{Y} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{bmatrix}.$$

Applying the direct Fourier logical transformation (10), we obtain

$$\mathbf{C} = \mathbf{A}_{\mathbf{2^3}}\mathbf{Y} = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{array}\right] \begin{bmatrix} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{bmatrix} = \begin{bmatrix} 0 \\ 7 \\ 6 \\ -12 \\ 5 \\ -10 \\ -8 \\ 19 \end{bmatrix} \begin{matrix} x_3 \\ x_2 \\ x_2x_3 \\ x_1 \\ x_1x_3 \\ x_1x_2 \\ x_1x_2x_3 \end{matrix}.$$

According to this example, the values of the coefficient vector $\mathbf{C}$ lie in the interval $-28 \le c_i \le 28$. Analyzing the structure of the matrices $\mathbf{A}_{2^n}$ and $\mathbf{A}_{2^n}^{-1}$, we find that their last rows contain the maximal number of unit elements. Moreover, the number of unit elements of like sign in the last row of the matrix $\mathbf{A}_{2^n}$ is 2^{n-1}. Since the maximal value taken by the elements of the matrix $\mathbf{Y}$ is $2^d - 1$ (d is the number of single-output Boolean functions realized), we find that the maximal absolute value of the coefficient (c_{2^n-1}) in the resultant matrix $\mathbf{C}$ is $abs(c_{2^n-1}) = 2^{n-1}(2^d - 1)$, where $abs(a)$ denotes the absolute value of a. To represent this coefficient in binary form with

regard for its sign, we require

$$N_{\mathbf{C}} = \left\lfloor \log_2(2^{n-1}(2^d - 1)) \right\rfloor + 2 = n + d \tag{12}$$

binary digits ($\lfloor x \rfloor$ is the largest integer not greater than x).

Large coefficients are critical for linear arithmetical polynomials. The reason for the large value of coefficients is primarily the large number of realized Boolean functions, which, in turn, arise as a result of garbage Boolean functions.

2.4. Investigation Problem

According to digital signal processing theory, modular arithmetical methods can be applied due to the advantages associated with the restricted number range of representation of intermediate results of transformations [13–15]. In this connection, let us study the application of modular transformation methods to weaken the "the problem of large coefficients" of arithmetical polynomials.

3. MODULAR ARITHMETICAL-LOGICAL FORMS

3.1. Polynomial Modular Arithmetical-Logical Forms

Unimodular arithmetic is the arithmetic of the ring of residues Z_m, where m is the value of the modulus [8–10]. The least nonnegative residue (or simply residue) of an integer N mod m is denoted by $|N|_m^+$.

Theorem 2. *If $m > Y_{\max}$, where $Y_{\max}$ is the maximal value of Y, then any cortege of Boolean functions can be represented by an arithmetical polynomial*

$$Y = \mu(X) = \left| \sum_{i=0}^{2^n-1} \psi_i x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_m^+, \tag{13}$$

where $\psi_i = |c_i|_m^+$ $(i = 0, 1, \dots, 2^n - 1)$.

The proof of Theorem 2 and design of algorithms for determining (13) are given in the Appendix.

Remark 1. In the general case, $m \geq 2^d$.

Definition 1. Expression (13) is called the representation of the Boolean function $f(X)$ based on the modular form of the arithmetical polynomial.

Two arithmetical polynomials $D(X)$ and $\mu(X)$ can be compared with the help of certain elementary Boolean functions (Table 2).

Table 2

$f(X)$	$D(X)$	$\mu(X)$
x_i	$1 - x_i$	$\lvert 1 + (m-1)x_i \rvert_m^+$
$x_1 \wedge x_2$	$x_1 x_2$	$x_1 x_2$
$x_1 \vee x_2$	$x_1 + x_2 - x_1 x_2$	$\lvert x_1 + x_2 + (m-1)x_1 x_2 \rvert_m^+$
$x_1 \oplus x_2$	$x_1 + x_2 - 2x_1 x_2$	$\lvert x_1 + x_2 + (m-2)x_1 x_2 \rvert_m^+$
$x_1 \wedge x_2$	$1 - x_1 x_2$	$\lvert 1 + (m-1)x_1 x_2 \rvert_m^+$
$x_1 \vee x_2$	$1 - x_1 - x_2 + x_1 x_2$	$\lvert 1 + (m-1)x_1 + (m-1)x_2 + x_1 x_2 \rvert_m^+$

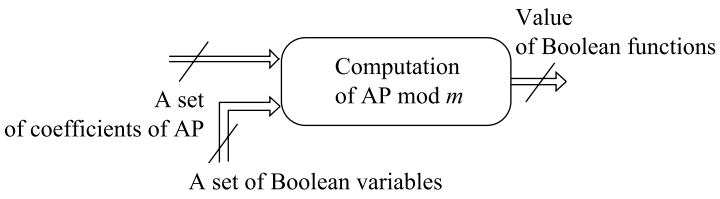

Fig. 1.

Realization of Boolean functions in unimodular arithmetic is illustrated by a block diagram in Fig. 1 (here and in what follows AP means arithmetical polynomial).

Corollary 1. *Coefficients of the arithmetical polynomial* $\mu(X)$ (13) *lie in the domain of nonnegative integers and their number range is equal to value of the modulus* m.

Corollary 2. *If two arithmetical polynomials* $D(X)$ (3) *and* $\mu(X)$ (13) *containing* K_1 *and* K_2 *terms, respectively, define a system of Boolean functions, then* $K_2 \le K_1$.

To explain Corollary 2, let us consider

Example 6. The system of Boolean functions (2), according to expression (3) (Example 2), corresponds to the arithmetical polynomial

$$Y = D(X) = 3 - 3x_1 - 3x_2 + 4x_1x_2.$$

Applying Theorem 2 to the general case, we obtain

$$Y = \mu(X) = |3 + (m-3)x_1 + (m-3)x_2 + 4x_1x_2|_m^+.$$

For $m = 4$,

$$\mu(X) = |3 + x_1 + x_2|_4^+.$$

Therefore Corollary 2 shows that the modular form of arithmetical polynomial (13) does not complicate the polynomial form of representation of systems of Boolean functions as judged by the indexes K_1 and K_2, and may be helpful in reducing the complexity of arithmetical polynomials by reducing the number of coefficients that are multiples of m. Consequently, the value of the modulus m can be chosen by its minimality criterion or by the criterion of minimality of K_2.

Lemma 1. *If a cortege of Boolean functions* (1) *is defined by a linear arithmetical polynomial* (7), *then for* $m > U_{\max}$, *the linear arithmetical polynomial can be expressed in modular form as*

$$U = \lambda(X) = \left|\omega_0 + \sum_{i=1}^{n} \omega_i x_i\right|_m^+ = |\omega_0 + \omega_1 x_1 + \ldots + \omega_n x_n|_m^+, \tag{14}$$

where $\omega_j = |d_j|_m^+$ $(j = 0, 1, \ldots, n)$.

The proof of Lemma 1 is given in the Appendix.

Remark 2. The parameter t of the operator $\Xi^t\{U\}$ does not change its value in passing from (7) to (14).

Definition 2. Expression (14) is called the representation of a Boolean function based on the modular form of a linear arithmetical polynomial.

Example 7. For the system of Boolean functions (8) defined by the linear arithmetical polynomial (9), the parameter t of the operator $\Xi^t\{U\}$ takes the maximal value $t_{\max} = 6$ and $U_{\max} = 36$. Choosing $m = 2^6 > 36$, we obtain

$$U = |20 + 60x_1 + 52x_2 + 16x_3|^+_{64}.$$

Let $x_1x_2x_3 = 011$. Consequently, $U = |88|^+_{64} = 24_{\text{dec}} = 011000$. Finally, we obtain

$$f_A(X) = \Xi^2\{011000\} = 0,$$
$$f_B(X) = \Xi^4\{011000\} = 1,$$
$$f_C(X) = \Xi^6\{011000\} = 0.$$

The operator $\Xi^t\{U\}$ and modular arithmetic are related by the expression

$$\Xi^t\{U\} = \left|\left\lfloor \frac{U}{2^t} \right\rfloor\right|^+_2.$$

Remark 3. If U is derived with the use of garbage Boolean functions of numbers greater than the maximal value $t_{\max}$ of the parameter t of the operator $\Xi^t\{U\}$, then the modulus m may be given the value $2^{t_{\max}}$. In this case, U in (14) must be replaced by $u = |U|^+_{2^{t_{\max}}}$, $u \le U$.

To explain Remark 3, let us consider

Example 8. For the system of Boolean functions

$$f_A(X) = \overline{x_1 \wedge x_2 \wedge x_3} = \Xi^3\{6 - x_1 - x_2 - x_3\},$$
$$f_B(X) = \overline{x_1 \oplus x_2 \oplus x_3} = \Xi^1\{1 + x_1 + x_2 + x_3\},$$

the linear arithmetical polynomial $L(X)$ is of the form

$$U = 2^3 f_B(X) + 2^0 f_A(X) = 14 + 7x_1 + 7x_2 + 7x_3.$$

Since $t_{\max} = 4$, according to Remark 3 we obtain $m = 16$ and

$$u = \lambda(X) = |14 + 7x_1 + 7x_2 + 7x_3|^+_{16}.$$

For $x_1x_2x_3 = 111$, we have $u = |35|^+_{16} = 2_{\text{dec}} = 0010$ and

$$f_A(X) = \Xi^3\{0010\} = 0,$$
$$f_B(X) = \Xi^4\{0010\} = 0,$$

i.e., by Remark 3, four digits are sufficient to represent U, instead of six as in (7) and (14).

Hence the modular form of the arithmetical polynomial (13) reduces the number range for computing the polynomial. Prior to estimating the number range, let us state the principles of realization of matrix transformations based on modular arithmetic.

3.2. Logical Number Theoretic Transforms in the Base $\mathbf{A}_{2^n}$

Theorem 3. *If a d-output Boolean function* $f(X)$ *is defined by two Fourier discrete logical transforms* (10) *and* (11) *and* $m > Y_{\max}$, *where* $Y_{\max}$ *is the maximal value of* Y, *then the following modular form of transforms holds:*

$$\mathbf{\Psi} = \mathbf{A}_2\mathbf{Y}(\bmod m), \tag{15}$$

$$\mathbf{Y} = \mathbf{A}_{2^n}^{-1}\mathbf{\Psi}(\bmod m), \tag{16}$$

where $\mathbf{A}_{2^n}$ *and* $\mathbf{A}_{2^n}^{-1}$ *are the matrices of direct and inverse arithmetical transformations, respectively, and* $\mathbf{Y}$ *and* $\mathbf{\Psi}$ *are the truth vector of the Boolean function* $f(X)$ *and vector of coefficients of the modular form of the arithmetical polynomial* $\mu(X)$ (13), *respectively. Here* $\bmod m$ *denotes that the arithmetical operations applied in the product of matrices* $\mathbf{A}_{2^n}$ *and* $\mathbf{A}_{2^n}^{-1}$ *by the column vector* $\mathbf{Y}$ *or* $\mathbf{\Psi}$ *are operations modulo* m.

To prove Theorem 3, we must take account of the one-one correspondence between the matrix forms (10) and (11) and polynomial forms (3) of representation of a system of Boolean functions [1]. Then the validity of (15) and (16) is implied by the validity of (13).

The pair of transforms thus obtained has many common properties with number theoretic transforms of digital signal processing methods [13–15].

Definition 3. Transforms (15) and (16) are called the modular form of direct and inverse matrix arithmetical transforms or logical number theoretic transforms, respectively.

Since $|-1|_m^+ = m-1$, expression (15) can be written in a different form as

$$\mathbf{\Psi} = \mathbf{M}_{2^n}\mathbf{Y}(\bmod m), \tag{17}$$

where $\mathbf{M}_{2^n} = |\mathbf{A}_{2^n}|_m^+$. The expression $|\mathbf{A}_{2^n}|_m^+$ denotes that the negative elements (units) of the matrix $\mathbf{A}_{2^n}$ are replaced by $m-1$.

Principles of realization of direct and inverse logical number theoretic transforms (15) and (16) can be explained with the help of graphs (Figs. 2 and 3), which are similar to the graphs of matrix transforms designed by Malyugin [1]. These graphs show that modular addition is used in both transforms, intermediate transformation results are nonnegative and not greater than the modulus m, and (3) computing resources to compute (15) and (16) can be reduced by methods of factorization of matrices $\mathbf{A}_{2^n}$ and $\mathbf{A}_{2^n}^{-1}$ [1]. Here transformation computation complexity may decrease $\xi = (3^n - 2^n)/(n2^{n-1})$ times [1].

Example 9. Let us illustrate the application of logical number theoretic transformations (15) and (16) to a two-output Boolean function (2) with truth matrix defined by Table 1 (cf. Example 2)

$$\mathbf{\Psi} = \mathbf{A}_{2^2}\mathbf{Y}(\bmod 2^2) = \left[\begin{array}{cc:cc} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ \hdashline -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 \end{array}\right] \begin{bmatrix} 3 \\ 0 \\ 0 \\ 1 \end{bmatrix} (\bmod 2^2) = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} \\ x_2 \\ x_1 \\ x_1x_2 \end{matrix},$$

$$\mathbf{Y} = \mathbf{A}_{2^2}^{-1}\mathbf{\Psi}(\bmod 2^2) = \left[\begin{array}{cc:cc} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \hdashline 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{array}\right] \begin{bmatrix} 3 \\ 1 \\ 1 \\ 0 \end{bmatrix} (\bmod 2^2) = \begin{bmatrix} 3 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

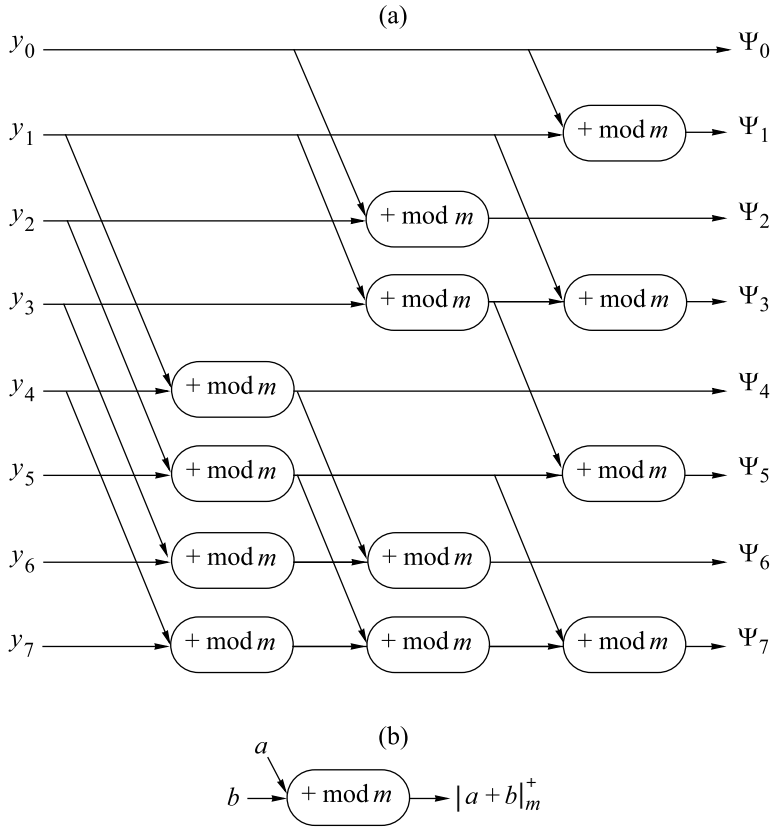
(a)
y_0
y_1
y_2
y_3
y_4
y_5
y_6
y_7
Ψ_0
Ψ_1
Ψ_2
Ψ_3
Ψ_4
Ψ_5
Ψ_6
Ψ_7
+ mod m
(b)
a
b
+ mod m
$|a+b|_m^+$

Fig. 2.

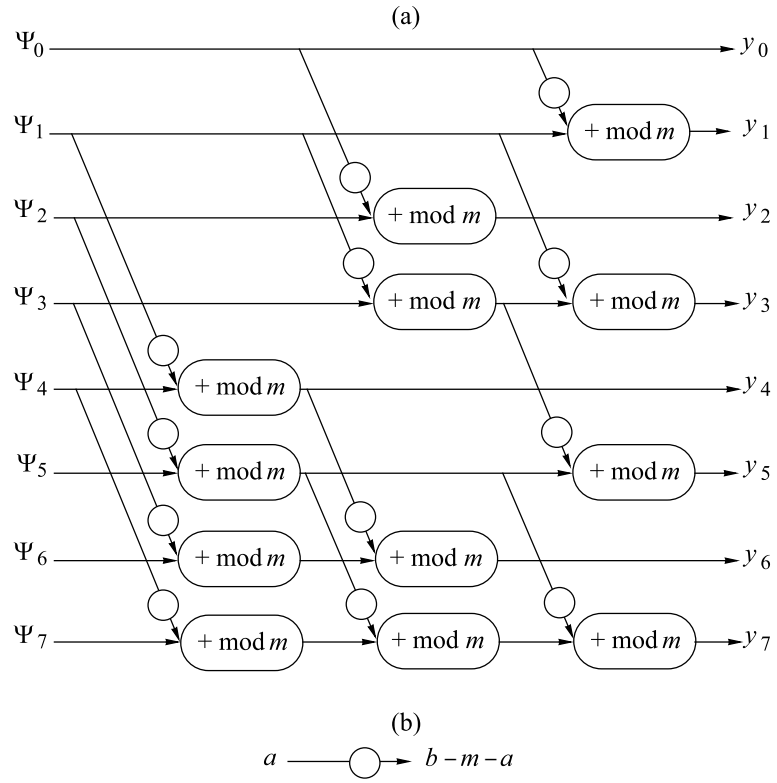
(a)
Ψ_0
Ψ_1
Ψ_2
Ψ_3
Ψ_4
Ψ_5
Ψ_6
Ψ_7
y_0
y_1
y_2
y_3
y_4
y_5
y_6
y_7
+ mod m
(b)
a
$b-m-a$

Fig. 3.

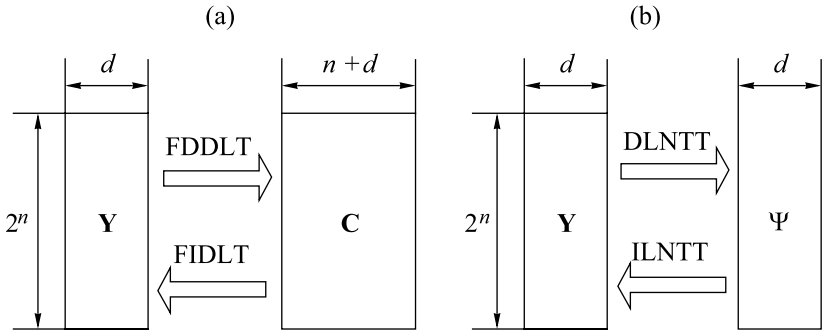

Fig. 4. FDDLT means Fourier direct discrete logical transformation, FIDLT means Fourier inverse discrete logical transformation, DLNTT means direct logical number theoretic transformation, and ILNTT means inverse logical number theoretic transformation.

Example 10. Applying the direct logical number theoretic transformation (17) for $m = 2^3$ to the three-output Boolean function of Example 5, we obtain

$$\boldsymbol{\Psi} = \mathbf{M}_{2^3}\mathbf{Y}(\bmod 2^3) = \left[\begin{array}{cccc:cccc} 1&0&0&0&0&0&0&0\\ 7&1&0&0&0&0&0&0\\ 7&0&1&0&0&0&0&0\\ 1&7&7&1&0&0&0&0\\ \hdashline 7&0&0&0&1&0&0&0\\ 1&7&0&0&7&1&0&0\\ 1&0&7&0&7&0&1&0\\ 7&1&1&7&1&7&7&1 \end{array}\right] \left[\begin{array}{c} 0\\7\\6\\1\\5\\2\\3\\7 \end{array}\right] (\bmod 2^3) = \left[\begin{array}{cc} 0 & \\ 7 & x_3\\ 6 & x_2\\ 4 & x_2x_3\\ 5 & x_1\\ 6 & x_1x_3\\ 0 & x_1x_2\\ 3 & x_1x_2x_3 \end{array}\right].$$

Hence the values of coefficients of the vector $\boldsymbol{\Psi}$ lie in the interval $0 \le \psi_i < 8$, $i = 0, 1, \ldots, 2^3-1$ (in the general case, $0 \le \psi_i < m$, $i = 0, 1, \ldots, 2^n - 1$), which is preserved in intermediate computations (compare with $-28 \le c_i \le 28$ in Example 5).

In analogy with logical Fourier discrete transforms, let us take the spectral matrix size as an estimate for the complexity of logical number theoretic transforms. Representation of the elements of the matrix $\boldsymbol{\Psi}$ requires $N_{\boldsymbol{\Psi}} = \lceil \log_2 m \rceil$ or $N_{\boldsymbol{\Psi}} = d$ for $m = 2^d$ binary digits ($\lceil x \rceil$ is the least integer equal to or greater than x) i.e.,

$$\frac{N_{\mathbf{C}}}{N_{\boldsymbol{\Psi}}} = \frac{n}{d} + 1 \tag{18}$$

times less than the number of digits $N_{\mathbf{C}}$ required for representing the elements of the matrix $\mathbf{C}$ (12).

Since $N_{\mathbf{C}}$ and $N_{\boldsymbol{\Psi}}$ are the maximal sizes (number of binary digits) of coefficients of arithmetic polynomials (3) and (13), estimate (18) also holds for the arithmetical polynomial (13).

Figure 4 shows the geometric interpretation of the gain resulting from the representation of the matrices $\mathbf{Y}$, $\mathbf{C}$, and $\boldsymbol{\Psi}$ (here matrix width denotes the number of binary digits required for representing the elements of column matrices of direct and inverse logical Fourier discrete transforms, and direct and inverse logical number theoretic transforms, respectively).

But this gain is preserved also for linear arithmetical polynomials, for which the number range of representation of coefficients can be only halved by moving computations to the domain of nonnegative numbers (in certain cases, Remark 3 may be applicable). Large value of the modulus m hinders further reduction of the number range.

4. MODULAR ARITHMETIC-LOGICAL FORMS BASED ON THE CHINESE REMAINDER THEOREM

4.1. Polynomial Modular Arithmetic-Logical Forms Based on the Chinese Remainder Theorem

In modelling real digital devices, coefficients of linear arithmetical polynomials may take absolute values greater than 2^{100} [11, 12]. Therefore, more sophisticated methods are required to reduce number ranges.

Let the modulus m for (13) and (14) be such that $m = \prod_{k=1}^{v} m_k$, where $\gcd(m_i, m_j) = 1$; $i, j = 1, \ldots, v$; $i \neq j$ (here and in what follows $\gcd(a, b)$ means the greatest common divisor of a and b). Then, according to the Chinese remainder theorem, Y can be uniquely expressed as a sequence $\{Y\} = (\phi_1, \phi_2, \ldots, \phi_v)$, where $\phi_k = |Y|^+_{m_k}$ $(k = 1, \ldots, v)$, $Y \in Z_m$ [8–10]. Applying the above approach to every residue ϕ_k $(k = 1, \ldots, v)$, we obtain

Theorem 4. *If $m > Y_{\max}$, where $m = \prod_{k=1}^{v} m_k$ and $\gcd(m_i, m_j) = 1$ $(i, j = 1, \ldots, v;\ i \neq j)$, then any cortege of Boolean functions can be uniquely represented by a system of modular forms of arithmetical polynomials*

$$\begin{cases} \phi_1 = \mu_1(X) = \left| \sum\limits_{i=0}^{2^n-1} \psi_{i,1} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|^+_{m_1} \\ \phi_2 = \mu_2(X) = \left| \sum\limits_{i=0}^{2^n-1} \psi_{i,2} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|^+_{m_2} \\ \vdots \\ \phi_v = \mu_v(X) = \left| \sum\limits_{i=0}^{2^n-1} \psi_{i,v} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|^+_{m_v}, \end{cases} \tag{19}$$

where $\psi_{i,k} = |c_i|^+_{m_k}$ $(i = 0, 1, \ldots, 2^n - 1;\ k = 1, \ldots, v)$.

The proof of Theorem 4 and the principles of design of algorithms (19) are given in the Appendix.

Definition 4. The system of arithmetical polynomials (19) is called the polynomial form of representation of Boolean functions based on the Chinese remainder theorem.

Remark 4. Modular forms (19) and (13) are related by the expressions $\{Y\} = (\phi_1, \phi_2, \ldots, \phi_v)$, $(\psi_{i,1}, \psi_{i,2}, \ldots, \psi_{i,v}) = \{\psi_i\} = |c_i|^+_m$ $(i = 0, 1, \ldots, 2^n - 1)$.

Corollaries 1 and 2 hold for every arithmetical polynomial of system (19) (here mod m must be replaced by mod m_j $(j = 1, \ldots, v)$).

Lemma 2. *If a linear arithmetic polynomial $L(X)$ (7) defines a cortege of Boolean functions (1), then for $m > U_{\max}$, where $m = \prod_{k=1}^{v} m_k$, and $\gcd(m_i, m_j) = 1$ $(i, j = 1, \ldots, v;\ i \neq j)$, the linear arithmetical polynomial can be expressed in modular form as*

$$\begin{cases} \phi_1 = \lambda_1(X) = |\omega_{0,1} + \omega_{1,1} x_1 + \cdots + \omega_{n,1} x_n|^+_{m_1} \\ \phi_2 = \lambda_2(X) = |\omega_{0,2} + \omega_{1,2} x_1 + \cdots + \omega_{n,2} x_n|^+_{m_2} \\ \vdots \\ \phi_v = \lambda_v(X) = |\omega_{0,v} + \omega_{1,v} x_1 + \cdots + \omega_{n,v} x_n|^+_{m_v}, \end{cases} \tag{20}$$

where $\omega_{j,k} = |d_j|^+_{m_k}$ $(j = 0, 1, \ldots, n;\ k = 1, 2, \ldots, v)$.

Validity of (20) is implied by the validity of (14) as applied to every number of modulus (20) separately and by the Chinese remainder theorem.

Definition 5. The system of arithmetical polynomials (20) is called the linear polynomial form of representation of Boolean functions based on the Chinese remainder theorem.

Remark 5. Modular forms (20) and (14) are related by the expressions $\{U\} = (\phi_1, \phi_2, \ldots, \phi_v)$; $(\omega_{j,1}, \omega_{j,2}, \ldots, \omega_{j,v}) = \{\omega_i\} = |d_j|_m^+$ $(j = 0, 1, \ldots, n)$.

Remark 3 is not applicable to (20).

For simplifying presentation, we shall not distinguish the numbers Y and U in what follows.

The system of equalities

$$\begin{cases} Y = \phi_1(\bmod m_1) \\ Y = \phi_2(\bmod m_2) \\ \vdots \\ Y = \phi_v(\bmod m_v) \end{cases}$$

is solved with the help of the Chinese remainder theorem. For this purpose, we use the expression

$$Y = \underset{k=1}{\overset{v}{\mathbf{CRT}}}\, \phi_k \ (\bmod m_k). \tag{21}$$

In modern interpretation of the Chinese remainder theorem, the formula

$$Y = \underset{k=1}{\overset{v}{\mathbf{CRT}}}\, \phi_k(\bmod m_k) = |\phi_1 B_1 + \phi_2 B_2 + \ldots + \phi_v B_v|_m^+, \tag{22}$$

is used for computing (21), where $B_k = q_k M m_k^{-1}$, q_k is determined from the congruence $q_k M m_k^{-1} \equiv 1(\bmod m_k)$ $(k = 1, \ldots, v)$ (here $a \equiv b(\bmod m_k)$ means a is congruent to b modulo m_k).

Example 11. Let us consider the linear arithmetical polynomial [11] $Y = L(X) = 8 + 137x_1 - 7x_2 + 129x_3 + 136x_4 + 64x_5$ realizing the system of Boolean functions

$$\begin{aligned} f_1(X) &= x_1 \oplus x_2 \oplus x_3, \\ f_2(X) &= x_1 \oplus \overline{x}_2 \oplus x_3, \\ f_3(X) &= x_5, \\ f_4(X) &= x_1 \oplus x_3 \oplus x_4. \end{aligned}$$

Moreover,

$$\begin{aligned} f_1(X) &= \Xi^1\{Y\}, \\ f_2(X) &= \Xi^4\{Y\}, \\ f_3(X) &= \Xi^7\{Y\}, \\ f_4(X) &= \Xi^8\{Y\}. \end{aligned}$$

Since $t_{\max} = 8$ and $Y_{\max} = 474$, we take $m_1 = 7$, $m_2 = 8$, and $m_3 = 9$ so that the condition $m = m_1 m_2 m_3 = 504 > 474$ is satisfied. Hence, according to (20),

$$\begin{cases} \lambda_1(X) = |1 + 4x_1 + 3x_3 + 3x_4 + x_5|_7^+ \\ \lambda_2(X) = |x_1 + x_2 + x_3|_8^+ \\ \lambda_3(X) = |8 + 2x_1 + 2x_2 + 3x_3 + x_4 + x_5|_9^+. \end{cases}$$

Let $x_1 x_2 x_3 x_4 x_5 = 10011$. Then

$$\begin{cases} \lambda_1(X) = 2 \\ \lambda_2(X) = 1 \\ \lambda_3(X) = 3. \end{cases}$$

Applying formula (22), since $B_1 = 288$, $B_2 = 441$, and $B_3 = 280$, we obtain

$$Y = |2 \times 288 + 1 \times 441 + 3 \times 280|^{+}_{504} = |1857|^{+}_{504} = 345_{\text{dec}} = 101011001.$$

Despite its classical form, formula (22) is not always convenient in practice, because a large number range is needed.

Theorem 5 ([16–18][1]). *A nonnegative integer* $Y < m = \prod_{k=1}^{v} m_k$ *represented by the residues* $\phi_1, \phi_2, \ldots, \phi_v$ *in a system of pairwise simple moduli* $m_1 < m_2 < \ldots < m_v$ *can be uniquely estimated by the recursion*

$$\begin{cases} h_1 = \phi_2, \\ h_2 = m_1 \left|\delta_1|\phi_1 - h_2|^{+}_{M_1}\right|^{+}_{M_1} + \phi_1 \\ h_3 = m_3 \left|\delta_2|\phi_3 - h_2|^{+}_{M_2}\right|^{+}_{M_2} + \phi_3 \\ h_4 = m_4 \left|\delta_3|\phi_4 - h_3|^{+}_{M_3}\right|^{+}_{M_3} + \phi_4 \\ \vdots \\ h_v = Y = m_v \left|\delta_{v-1}|\phi_v - h_{v-1}|^{+}_{M_{v-1}}|^{+}_{M_{v-1}}\right|^{+}_{M_{v-1}} + \phi_v, \end{cases} \tag{23}$$

where $M_1 = m_2$, $M_i = \prod_{j=1}^{i} m_j$, $\delta_1 = \left|(M_1 - m_1)^{-1}\right|^{+}_{M_i}$, *and* $\delta_i = \left|(M_i - m_{i+1})^{-1}\right|^{+}_{M_i}$.

The validity of (23) is demonstrated with the help of the generalization of Theorem 5 to two moduli [17, 18] via step-by-step passage to composite moduli and enlarged number range.

To derive Y with the help of (23), we require v steps. Here the intermediate results do not escape from the range of M_v. If computations are parallelized, there is a variant of Theorem 5 for implementing a transformation in $\lceil \log_2 v \rceil$ steps [17, 18].

Figure 5 shows a simple block diagram explaining the principle of realization of Boolean functions by modular forms of arithmetical polynomials based on the Chinese remainder theorem.

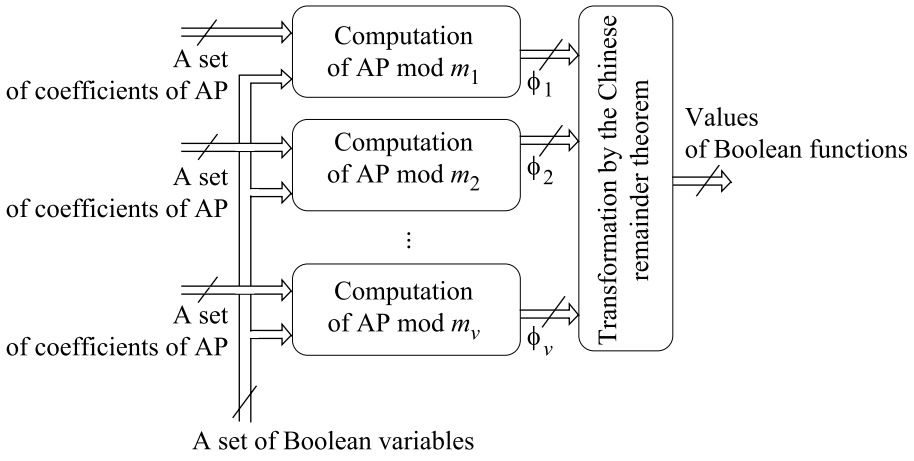

Fig. 5.

[1] After the publication of [16], similar results were reported in [19].

4.2. Number Theoretic Transformation in Base $\mathbf{A}_{2^n}$ by the Chinese Remainder Theorem

Lemma 3. *If two logical number theoretic transforms* (15) *and* (16) *are given for a d-output Boolean function $f(X)$ and $m > Y_{\max}$, where $m = \prod_{k=1}^{v} m_k$ and $\gcd(m_i, m_j) = 1$ $(i, j = 1, \ldots, v;$ $i \neq j)$, then transforms can be expressed in modular arithmetic-logical form as*

$$\begin{cases} \mathbf{\Psi}_1 = \mathbf{A}_{2^n}\mathbf{\Phi}_1 \,(\mathrm{mod}\, m_1) \\ \mathbf{\Psi}_2 = \mathbf{A}_{2^n}\mathbf{\Phi}_2 \,(\mathrm{mod}\, m_2) \\ \vdots \\ \mathbf{\Psi}_v = \mathbf{A}_{2^n}\mathbf{\Phi}_v \,(\mathrm{mod}\, m_v); \end{cases} \tag{24}$$

$$\begin{cases} \mathbf{\Phi}_1 = \mathbf{A}_{2^n}^{-1}\mathbf{\Psi}_1 \,(\mathrm{mod}\, m_1) \\ \mathbf{\Phi}_2 = \mathbf{A}_{2^n}^{-1}\mathbf{\Psi}_2(\ \mathrm{mod}\, m_2) \\ \vdots \\ \mathbf{\Phi}_v = \mathbf{A}_{2^n}^{-1}\mathbf{\Psi}_v(\ \mathrm{mod}\, m_v), \end{cases} \tag{25}$$

where $\mathbf{A}_{2^n}$ and $\mathbf{A}_{2^n}^{-1}$ are the matrices of direct and inverse arithmetical transforms, respectively, and $\mathbf{\Phi}_k = \left[\phi_k^{(0)}, \phi_k^{(1)}, \ldots, \phi_k^{(2^n-1)}\right]^{\mathrm{T}}$, $\phi_k^{(r)} = \left|Y^{(i)}\right|_{m_k}^{+}$, $k = 1, \ldots, v$; $\mathbf{\Psi}_k = [\psi_{0,k}, \psi_{1,k}, \ldots, \psi_{2^n-1,k}]^{\mathrm{T}}$ $(k = 1, \ldots, v)$.

The validity of (24) and (25) is implied by the one-one correspondence between matrix forms (10) and (11) and polynomial form (3) of representation of Boolean functions [1] and the proof of the polynomial representation (19) based on the Chinese remainder theorem.

Definition 6. Systems of matrix transforms (24) and (25) are said to be the logical number theoretic transforms based on the Chinese remainder theorem.

Remark 6. The logical number theoretic transforms (24) and (25) based on the Chinese remainder theorem are related to the logical number theoretic transforms (15) and (16) by the relations $\mathbf{\Psi} = \underset{k=1}{\overset{v}{\mathbf{CRT}}}\, \mathbf{\Psi}(\,\mathrm{mod}\, m_k)$ and $\mathbf{Y} = \underset{k=1}{\overset{v}{\mathbf{CRT}}}\, \mathbf{\Phi}_k(\,\mathrm{mod}\, m_k)$.

Example 12. Let us consider two logical number theoretic transforms based on the Chinese remainder theorem that are applicable under the conditions of Examples 5 and 10. Let us choose the moduli $m_1 = 3$ and $m_2 = 5$ such that the number range is $M = m_1 m_2 = 15 \geq 2^3$. Then

$$|\mathbf{Y}|_3^+ = \left|\begin{bmatrix} 0&0&0 \\ 1&1&1 \\ 1&1&0 \\ 0&0&1 \\ 1&0&1 \\ 0&1&0 \\ 0&1&1 \\ 1&1&1 \end{bmatrix}\right|_3^+ = \left|\begin{bmatrix} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{bmatrix}\right|_3^+ = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 2 \\ 2 \\ 0 \\ 1 \end{bmatrix}; \qquad |\mathbf{Y}|_5^+ = \left|\begin{bmatrix} 0&0&0 \\ 1&1&1 \\ 1&1&0 \\ 0&0&1 \\ 1&0&1 \\ 0&1&0 \\ 0&1&1 \\ 1&1&1 \end{bmatrix}\right|_5^+ = \left|\begin{bmatrix} 0 \\ 7 \\ 6 \\ 1 \\ 5 \\ 2 \\ 3 \\ 7 \end{bmatrix}\right|_5^+ = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 1 \\ 0 \\ 2 \\ 3 \\ 2 \end{bmatrix}.$$

Transform (24) takes the form

$$\mathbf{\Psi}_1 = \mathbf{A}_{2^3}\mathbf{\Phi}_1(\,\mathrm{mod}\, 3) = \mathbf{A}_{2^3}[0\,1\,0\,1\,2\,2\,0\,1]^{\mathrm{T}}(\,\mathrm{mod}\, 3) = [0\,1\,0\,0\,2\,2\,1\,1]^{\mathrm{T}},$$
$$\mathbf{\Psi}_2 = \mathbf{A}_{2^3}\mathbf{\Phi}_2(\,\mathrm{mod}\, 5) = \mathbf{A}_{2^3}[0\,2\,1\,1\,0\,2\,3\,2]^{\mathrm{T}}(\,\mathrm{mod}\, 5) = [0\,2\,1\,3\,0\,0\,2\,4]^{\mathrm{T}},$$

Using (23), we can find Y for the case of two moduli [17, 18, 20]:

$$Y^{(i)} = m_1 \left|\delta_1 |\phi_1^{(i)} - \phi_2^{(i)}|_{m_2}^{+}\right|_{m_2}^{+} + \phi_1^{(i)} = 2\left|2|\phi_1^{(i)} - \phi_2^{(i)}|_5^{+}\right|_5^{+} + \phi_1^{(i)}, \quad i = 0, 1, \ldots, 2^n - 1.$$

Thus we obtain

$$Y^{(0)} = 2\left|2|0-0|_5^{+}\right|_5^{+} + 0 = 0, \quad Y^{(4)} = 2\left|2|2-0|_5^{+}\right|_5^{+} + 2 = 5,$$
$$Y^{(1)} = 2\left|2|1-2|_5^{+}\right|_5^{+} + 1 = 7, \quad Y^{(5)} = 2\left|2|2-2|_5^{+}\right|_5^{+} + 2 = 2,$$
$$Y^{(2)} = 2\left|2|0-1|_5^{+}\right|_5^{+} + 0 = 6, \quad Y^{(6)} = 2\left|2|0-3|_5^{+}\right|_5^{+} + 0 = 3,$$
$$Y^{(3)} = 2\left|2|1-1|_5^{+}\right|_5^{+} + 1 = 1, \quad Y^{(7)} = 2\left|2|1-2|_5^{+}\right|_5^{+} + 1 = 7.$$

Hence

$$\mathbf{\Psi} = \overset{2}{\underset{k=1}{\mathbf{CRT}}}\left[\psi_{0,k}, \psi_{1,k}, \ldots, \psi_{2^3-1,k}\right]^{\mathrm{T}} (\bmod m_k) = [0\,7\,6\,8\,5\,0\,2\,9]^{\mathrm{T}},$$
$$\mathbf{Y} = \overset{2}{\underset{k=1}{\mathbf{CRT}}}[\phi_{0,k}, \phi_{1,k}, \ldots, \phi_{2^3-1,k}]^{\mathrm{T}} (\bmod m_k) = [0\,7\,6\,1\,5\,2\,3\,7]^{\mathrm{T}}.$$

The graphs of matrix transforms for every k of modulus m_k of the logical number theoretic transform based on the Chinese remainder theorem are similar to the graphs in Figs. 2 and 3. Consequently, the methods of factorization of the matrices $\mathbf{A}_{2^n}$ and $\mathbf{A}_{2^n}^{-1}$ (fast conjunctive transformation methods) applicable to the Fourier logical discrete transforms [1] and logical number theoretic transforms are also applicable to logical number theoretic transforms based on the Chinese remainder theorem.

Figure 6 shows the geometric interpretation of the logical number theoretic transform based on the Chinese remainder theorem and its relationship with the Fourier logical discrete transforms [1] and logical number theoretic transforms.

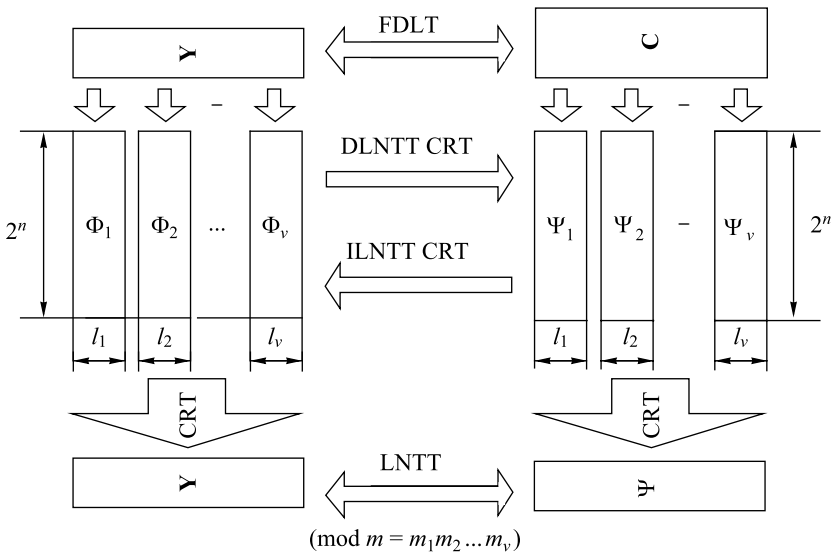

Fig. 6. FDLT means Fourier discrete logical transformation, DLNTT CRT means direct logical number theoretic transformation based on the Chinese remainder theorem, ILNTT CRT means inverse logical number theoretic transformation based on the Chinese remainder theorem, LNTT means logical number theoretic transformation, and CRT means the Chinese remainder theorem.

Table 3

m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}	m_{16}	m_{17}
67	71	73	79	83	89	91	97	101	103	107	109	111	113	115	127	128

According to this diagram, the logical number theoretic transform based on the Chinese remainder theorem is reduced to decomposing each of the matrices $\mathbf{Y}$ and $\mathbf{C}$ into v matrices of lesser "width"—$\ell_1, \ell_2, \ldots, \ell_v$, where $\ell_k = \lceil \log_2 m_k \rceil$, such that the transform for each of the matrices $\mathbf{\Psi}_k$ or $\mathbf{\Phi}_k$ $(k = 1, 2, \ldots, v)$ is simplified separately. These results are restored with the help of the Chinese remainder theorem. Moreover, the spectrum $\mathbf{\Psi}$ is a matrix of logical number theoretic transforms modulo $m = \prod_{k=1}^{v} m_k$.

Assuming that all moduli of the logical number theoretic transform based the Chinese remainder theorem are transformed concurrently in parallel, we find that the transformation of the logical number theoretic transform based the Chinese remainder theorem for the largest modulus m_v has the maximal complexity. The maximal "width" (number of binary digits needed for representing the elements of the matrix) corresponds to the matrix $\mathbf{\Psi}_v$ of the largest modulus m_v and is equal to $\ell_v = \lceil \log_2 m_v \rceil$ binary digits. Since ℓ_v is not greater than six or seven for most of practical problems, the gain compared to Fourier logical discrete transforms and logical number theoretic transforms is

$$\frac{N_{\mathbf{C}}}{\ell_v} \approx \frac{n+d}{6 \div 7} \quad \text{and} \quad \frac{N_{\mathbf{\Psi}}}{\ell_v} \approx \frac{d}{6 \div 7} \text{ times, respectively.}$$

For example, for $n = d = 40$ and $m_v = 128$, the gains are 11.4 and 5.7 times, respectively. For restoring the final result by (23), we require v iterations. For example, a set of 17 moduli (Table 3) produces a representation of coefficients for the arithmetical polynomial (3) or (7) with upper bound 2^{105} (rounded to the least degree of 2).

These estimates also hold for modular forms of polynomial transforms based on the Chinese remainder theorem. Furthermore, for comparing with a linear arithmetical polynomial, we must apply the ratio $\dfrac{d^*}{6 \div 7}$, where d^* is the total number (including garbage) Boolean functions or the maximal number of digits required for representing the polynomial coefficients. For $d^* = 2^{100}$, the gain in the number of binary digits of numbers in a transformation channel (modulus) is 14.3 fold.

5. CONCLUSIONS

The classification of modular forms of representation for Boolean functions shown in Fig. 7 demonstrates that logical number theoretic transforms are used for the first time along with modular forms of (linear and nonlinear) arithmetical polynomials.

The main advantage of modular forms of representation for Boolean functions based on unimodular arithmetic is the reduction in the number range of representation for the results of intermediate transformations. This advantage results from the restriction imposed on the values of coefficients (and their sums) of arithmetical polynomials by the value of the modulus of transformation (mostly for nonlinear arithmetical polynomials) and from the translation of coefficients to the domain of positive numbers. But for linear arithmetical polynomials, unimodular arithmetic is not of any advantage. Radical reduction in the number range yields modular forms of representation of Boolean functions based on the Chinese remainder theorem. These modular forms are most effective for realizing large systems of Boolean functions (large values of d) for large values of coefficients of arithmetical polynomials ($2^{100} \div 2^{1000}$ or higher). The Chinese remainder theorem for estimating a number from its residues is modified such that the number range is minimal. But these advantages

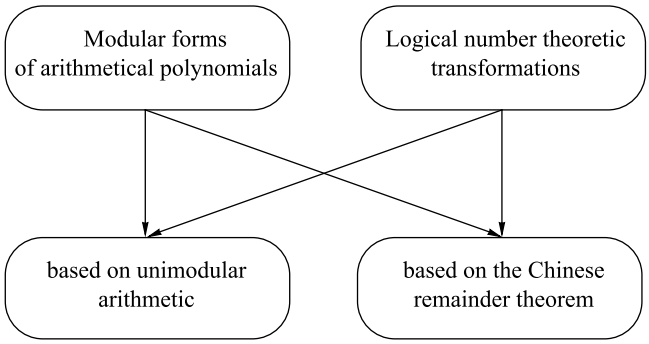

Fig. 7.

hold only if groups of or all transformation moduli are computed concurrently. Although this narrows the range of application of these forms, digital signal processing techniques and devices can be used for realizing logical computations.

These methods may be expected to be more effective in realizing functions of the algebra of multi-valued logic if multiplication operation is accelerated. These methods can be regarded as methods of computing k-valued logical functions for arbitrary k-program or hardware devices with logic of different values (values of moduli). Further research in this field may cover the application of modular forms for controlling errors in logical computations [21].

Our main results were tested and reported in International conferences [18, 22–26].

APPENDIX

Proof of Theorem 2. Let us consider two variants of proofs, which define two variants of the algorithm for determining (13).

(a) For a given arithmetical polynomial $D(x)$ (3), according to the theory of congruences [27],

$$|Y|_m^+ = \left|\sum_{i=0}^{2^n-1} c_i x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}\right|_m^+ = \left|\sum_{i=0}^{2^n-1} |c_i|_m^+ \, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}\right|_m^+. \tag{A.1}$$

But if $Y < m$ (Y is nonnegative), then $|Y|_m^+ = Y$. Consequently, for $Y < m$,

$$Y = \left|\sum_{i=0}^{2^n-1} |c_i|_m^+ \, x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}\right|_m^+ = \left|\sum_{i=0}^{2^n-1} \psi_i x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}\right|_m^+, \quad \text{where} \quad \psi_i = |c_i|_m^+. \tag{A.2}$$

This completes the proof. By this proof, the algorithm for determining (13) consists of

Step 1. Construction of (3) by algorithm 1 and

Step 2. Determination of (13) from expression (A.2).

(b) Since $f_j(X) \in \{0, 1\}$ and $m > Y$, reasoning as before, we can express polynomial (4) as

$$f_j(X) = \mu'_j(X) = \left|\sum_{i=0}^{2^n-1} \psi'_{j,i} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}\right|_m^+, \quad j = 1, \dots, d, \tag{A.3}$$

where $\psi'_{j,i} = |r_{j,i}|_m^+$ ($j = 1, \dots, d$; $i = 0, 1, \dots, 2^n - 1$). Now expressing (5) and (6) mod m, we obtain

$$\mu''_j(X) = \sum_{i=0}^{2^n-1} \psi''_{j,i} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}, \quad j = 1, \dots, d, \tag{A.4}$$

where $\psi''_{j,i} = |\psi'_{j,i} b_j|^+_m$, $b_j = |2^{j-1}|^+_m$ $(j = 1, \ldots, d;\ i = 0, 1, \ldots, 2^n - 1)$, and

$$Y = \mu(X) = \left| \sum_{i=0}^{2^n-1} \left| \sum_{j=1}^{d} \psi''_{j,i} \right|^+_m x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|^+_m = \left| \sum_{i=0}^{2^n-1} \psi_i x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|^+_m . \tag{A.5}$$

This completes the proof of the theorem.

Therefore, the second variant of the algorithm for determining (13) consists of three steps (A.3), (A.4), and (A.5).

Proof of Lemma 1. Let us consider the arithmetical polynomial

$$U = \left| \omega_0^* + \sum_{i=1}^{s} \omega_i^* z_i \right|^+_m = |\omega_0^* + \omega_1^* z_1 + \ldots + \omega_n^* z_n|^+_m .$$

Substituting $\omega_0^* = \omega_0$, $\omega_i^* = \omega_i$, $z_i = x_i$, and $s = n$, we obtain polynomial (14). Let $U = Y$, $\omega_0^* = \psi_0$, $\omega_i^* = \psi_i$, $z_i = x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$, and $s = 2^n - 1$. Therefore, (14) is demonstrated by repeating step (a) in the proof of Theorem 2.

Proof of Theorem 4. As in the proof of Theorem 2, let us consider two variants for the proofs defining two variants of algorithm for determining (19).

(a) Since (A.1) holds for any modulus $m > 1$, we find that v expressions of (A.1) hold for different values of the modulus $m > 1$:

$$\begin{aligned}
|Y|^+_{m_1} &= \left| \sum_{i=0}^{2^n-1} c_i x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|^+_{m_1} = \left| \sum_{i=0}^{2^n-1} |c_i|^+_{m_1} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|^+_{m_1}, \\
|Y|^+_{m_2} &= \left| \sum_{i=0}^{2^n-1} c_i x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|^+_{m_2} = \left| \sum_{i=0}^{2^n-1} |c_i|^+_{m_2} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|^+_{m_2}, \\
&\vdots \\
|Y|^+_{m_v} &= \left| \sum_{i=0}^{2^n-1} c_i x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|^+_{m_v} = \left| \sum_{i=0}^{2^n-1} |c_i|^+_{m_v} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \right|^+_{m_v} .
\end{aligned} \tag{A.6}$$

By the Chinese remainder theorem, the system of residues

$$\phi_1 = |Y|^+_{m_1},\ \phi_2 = |Y|^+_{m_2},\ \ldots,\ \phi_v = |Y|^+_{m_v}$$

has a unique solution Y if

$$Y_{\max} < \prod_{k=1}^{v} m_k; \quad \gcd(m_i, m_j) = 1, \quad i, j = 1, \ldots, v;\ i \neq j.$$

Since these conditions are defined by Theorem 4, we find that (19) has also a unique solution Y.

The completes the proof of the theorem.

According to this proof, (19) is constructed in two steps:

Step 1. Construction of (3) by algorithm 1.

Step 2. Construction of system (19) using (A.6).

(b) Using (A.3), we construct a system of arithmetical polynomials for v given moduli satisfying the conditions for representation (19) to be unique:

$$P_j(X) = \begin{cases} \mu'_{j,1}(X) = \left| \sum\limits_{i=0}^{2^n-1} \psi'_{j,i,1} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_{m_1}^{+} \\ \mu'_{j,2}(X) = \left| \sum\limits_{i=0}^{2^n-1} \psi'_{j,i,2} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_{m_2}^{+} \\ \vdots \\ \mu'_{j,v}(X) = \left| \sum\limits_{i=0}^{2^n-1} \psi'_{j,i,v} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \right|_{m_v}^{+} \end{cases}, \tag{A.7}$$

where $\psi'_{j,i,k} = |r_{j,i}|^{+}_{m_k}$ $(j = 1, \dots, d;\ i = 0, 1, \dots, 2^n - 1;\ k = 1, \dots, v)$.

Similarly, using (A.4) we obtain

$$\begin{cases} \mu''_{j,1}(X) = \sum\limits_{i=0}^{2^n-1} \psi''_{j,i,1} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \\ \mu''_{j,2}(X) = \sum\limits_{i=0}^{2^n-1} \psi''_{j,i,2} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n} \\ \vdots \\ \mu''_{j,v}(X) = \sum\limits_{i=0}^{2^n-1} \psi''_{j,i,v} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}, \end{cases} \tag{A.8}$$

where $\psi''_{j,i,k} = |\psi'_{j,i,k} b_{j,k}|^{+}_{m_k}$, $b_{j,1} = |2^{j-1}|^{+}_{m_1}, b_{j,2} = |2^{j-1}|^{+}_{m_2}, \dots, b_{j,s} = |2^{j-1}|^{+}_{m_s}$ $(j = 1, \dots, d;\ i = 0, 1, \dots, 2^n - 1;\ k = 1, \dots, v)$.

Finally, generalizing (A.4) for the moduli $m_1, m_2, \dots, m_v$, we obtain (19) from the expressions

$$\psi_{i,k} = \left| \sum_{j=1}^{d} \psi''_{j,i,k} \right|_{m_k}^{+} \qquad (i = 0, 1 \dots, 2^n - 1;\ k = 1, \dots, v). \tag{A.9}$$

This completes the proof of the theorem.

Algorithm for finding (19) consists of three steps for constructing (A.7) and (A.8), and (19) with (A.9).

The first variants of realization of algorithms for generating arithmetical polynomials (13) and (19) are simple, whereas the second variants consist of forming the respective arithmetical polynomials such that transforms remain modular in all realization stages in order to "confine" intermediate results within the limits a given number range.

REFERENCES

1. Malyugin, V.D., *Parallel'nye logicheskie vychisleniya posredstvom arifmeticheskikh polinomov* (Parallel Logical Computation through Arithmetical Polynomials), Moscow: Nauka, 1997.

2. Malyugin, V.D., Realization of Boolean Functions by Arithmetical Polynomials, *Avtom. Telemekh.*, 1982, no. 4, pp. 84–93.

3. Malyugin, V.D., Realization of a Cortege of Boolean Functions by Linear Arithmetical Polynomials, *Avtom. Telemekh.*, 1984, no. 2, pp. 114–121.

4. Thornton, M.A., Dreschler, R., and Miller D.M., *Spectral Techniques in VLSI CAD*, London: Kluwer Academic, 2002.

5. Heidtmann, K.D., Arithmetic Spectrum Applied to Fault Detection for Combinational Natworks, *IEEE Trans. Comput.*, 1991, vol. 40, no. 3, pp. 320–324.

6. Shmerko, V.P., Design of Arithmetical Forms of Boolean Functions by Fourier Transformation, *Avtom. Telemekh.*, 1989, no. 5, pp. 134–142.

7. Kukharev, G.A., Shmerko, V.P., and Zaitseva, E.N., *Algoritmy i sistolicheskie protsessory dlya obrabotki mnogoznachnykh dannykh* (Algorithms and Systolic Processors for Processing Multi-Valued Data), Minsk: Nauka i Tekhnika, 1990.

8. Akritas A.G., *Elements of Computer Algebra*, New York: Wiley, 1989. Translated under the title *Osnovy komp'yuternoi algebry s prilozheniyami*, Moscow: Mir, 1994.

9. Knuth, D.E., *The Art of Computer Programming, vol. 2, Seminumerical Algorithms*, Reading: Addison-Wesley, 1969. Translated under the title *Iskusstvo programmirovaniya, tom 2, poluchislennye algoritmy*, Moscow: Williams, 2000.

10. Amerbaev, V.M., *Teoreticheskie osnovy mashinnoi arifmetiki* (Theoretical Principles of Machine Arithmetic), Alma-Ata: Nauka, 1976.

11. Yanushkevich, S., Shmerko, V., and Dziurzanski, P., Word Level Decision Diagrams Upon the Conditions of Linearity, *IEEE Trans. Comput. Design Integrated Syst.*, 2001, vol. XX, no. Y, pp. 1–40.

12. Shmerko, V.P., Malyugin's Theorem: A New Concept in Logical Control, VLSI Design, and Data Structures for New Technologies, *Avtom. Telemekh.*, 2004, no. 6.

13. Soderstrand, M.A., Jenkins, W.K., Jullien, G.A., and Tailor, F.J., *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*, New York: IEEE Press, 1986.

14. Maclellan, J.H. and Rader, C.M., *Number Theory in Digital Signal Processing*, Englewood Cliffs: Prentice Hall, 1979. Translated under the title *Primenenie teorii chisel v tsifrovoi obrabotke signalov*, Moscow: Radio i Svyaz', 1983.

15. Kravchenko, V.F. and Krot, A.M., Methods and Microelectronic Tools for Digital Signal Filtration and Imaging based on Number Theoretic Transforms, *Zarubezh. Radioelektr. Usp. Sovrem. Radioelektr.*, 1997, no. 6, pp. 3–31.

16. Chervyakov, N.I., Korshunov, O.E., and Fin'ko, O.A., Converter of Codes from a System of Residue Classes into Positional Codes, Patent no. 1388996, *Byul. Izobret.*, 1988, no. 14, p.167.

17. Fin'ko, O.A., Reproduction of a Number in a System of Residual Classes with Minimal Number of Bases, *Elektron. Modelir.*, 1998, vol. 20, no. 3, pp. 56–61.

18. Fin'ko, O.A., Different Variants of the Chinese Remainder Theorem for Technical Realization, *Proc. Int. Congr. "Mathematics in the XXI Century. Role of MMF-NGU in Science, Education, and Business,"* Novosibirsk, 2003.

19. Naudin, P. and Quitte, C., *Algorithmique algebrique*, Paris: Masson, 1992. Translated under the title *Algebraicheskaya algoritmika*, Moscow: Mir, 1999.

20. Chervyakov, N.I., Korshunov, O.E., and Fin'ko, O.A., A Converter of Codes from a System of Residue Classes into Positional Codes, Patent no. 1343553, *Byul. Izobret.*, 1987. no. 37, p. 288.

21. Fin'ko, O.A., Control and Reconfiguration of Analog-Digital Devices Operating in a System of Residue Classes, *Elektron. Modelir.*, 2000, vol. 22, no. 4, pp. 92–103.

22. Fin'ko, O.A., Methods of Problem-Oriented Representation and Data Processing in Resources of the Hardware Support of Intellectual Systems, *Proc. IEEE Conf. Artificial Intelligence Syst.* (*AIS'02*), Divnomorskoe, September 5–10, 2002, pp. 453–454.

23. Fin'ko, O.A., Superparallel Logical Computations by Modular Arithmetic Methods, *Proc. Int. Conf. "Artificial Intelligent Systems"* (IEEE AIS'02) and "Intelligent CAD" (CAD-2002), Moscow: Nauka, 2002, pp. 448–455.

24. Fin'ko, O.A., Modular Forms of Arithmetical Polynomials for Realization of Systems of Boolean Functions, *Proc. Int. Conf. "Artificial Intelligent Systems" (IEEE AIS'03) and "Intelligent CAD"* (CAD-2003), Moscow: Nauka, 2003, pp. 611–619.

25. Fin'ko, O.A., Parallel Logical Computations using Redundant Number Representations, *Proc. II Int. Conf. "Identification of Systems and Control Problems"* (SICPRO'03), Moscow: Inst. Problem Upravlen., 2003, pp. 1716–1728.

26. Fin'ko, O.A., Logical Computations via Number Theoretic Transforms *Proc. II Int. Conf. Control Problems*, Moscow: Inst. Problem Upravlen., 2003.

27. Bukhshtab, A.A. *Teoriya chisel* (Number Theory), Moscow: Prosveshchnie, 1966.

This paper was recommended for publication by P.P. Parkhomenko, a member of the Editorial Board