Linear Logic Polynomials

V.D. Malyugin[1]

Abstract - **This paper considers a class of linear arithmetic polynomials, generalized disjunctive forms, generalized Zhegalkin polynomials and their mixed forms for representation of switching functions. Their time complexity and complexity of testing are estimated. Applications of linear arithmetic polynomials in interpretation of commutators are discussed.**

Keywords - **Switching functions, arithmetic polynomials, testing of digital devices.**

I. Linear Arithmetic Polynomials

Linear arithmetic polynomials [1] are expressions of the form

$$P(x) = a_0 + \sum_{i=1}^{n} a_i x_i, \text{ with } x_i \in \{0, 1\};$$

a_0, $a_i \in Z$, where Z is the set of integers.

In general, a linear polynomial is an expression of the form

$a_0 \diagup a_1 x_1 \diagup a_2 x_2 \diagup \ldots \diagup a_n x_n$,

where $a \in \{0,1\}$, and symbol $\diagup \in \{+, \vee, \oplus\}$.

It is obvious that serial realization of this polynomial requires n clock intervals. For practical applications, it is a crucial question, whether it is possible to realize this polynomial with a smaller number of clocks

We represent each terms $a_i x_i$ of the polynomial as a system (cortege) consisting of functions on the set $\{x_i, 0, 1\}$, $i=1\ldots k$. Thus, $a_i x_i = \underset{j}{*}\left(a_{ij} x_i\right)$, where a_{ij} - is a binary representation of the number a_i. For example,

$5x_2 = 101\, x_2 = x_2 * 0 * x_2$,
$8x_3 = 1000x_3 = x_3 * 0 * 0 * 0$.

Two corgetes are orthogonal if their componentwise product is the zero cortege.

The sum of mutually orthogonal corteges is determined by the combination of non-zero coordinates of the given corteges. For example,

$5\, x_2 + 8\, x_3 = x_3 * x_2 * 0 * x_2$.

We represent the coefficients of a given polynomial as a binary matrix. Due to that, determination of the values of the polynomial reduces to the addition of several coefficients. It can be shown that union of all mutually orthogonal polynomials is a set of at most n polynomials.

It follows that determination of the sum of a linear polynomial of n variables requires s clocks, where $0 \le s \le n$. Thus, the procedure for realization requires less that n clocks.

Consider a procedure for testing of a linear polynomial realizing a given system of m functions. In that case, we reduce the testing of realization of m functions to testing of a correct realization of a uniquely determined polynomial generating these functions. Testing is realized by exciting the inputs of the device under testing by code combinations of input variables and by comparison of the results with given etalon values.

We will explain the basic principle of this approach by the example of a cortege of functions Fm to which corresponds a linear polynomial of the form

$$P_a(x) = a_0 + \sum_{i=1}^{n} a_i x_i .$$

We consider that a polynomial is realized correctly, if its coefficients take correct values, the operations are performed correctly, including both the arithmetic addition and multiplication of coefficients by a binary variable.

In other words, a model for an unreliable polynomial is a polynomial that differs from the given polynomial in coefficients for the corresponding variables, i.e., the polynomial

$$P_z(x) = z_0 + \sum_{i} z_i x_i .$$

In this ways, testing of a given polynomial reduces to the procedure of finding the real values of the coefficients of the realized polynomial for $z_0, z_1, \ldots, z_n$. It remains to check the $a_i = z_i$ for all i.

To do that, we assign to variables $x_1, \ldots, x_n$ the values selected such that from the polynomial $P_z(x)$ we get n+1 joint equations, whose solutions produce the required coefficients. It can be shown that this goal is achieved by selecting z_0 and

[1] Institute of Control, Russian Academy of Sciences, Profsoysnaya 65, Moscow, Russia

variables with weight coefficients equal to 1 as shown in the following table

$x_1x_2 \ldots x_n$	$P(x)$
00…0	z_0
10…0	z_0+z_1
01…0	z_0+z_2
…….	……
00…1	z_0+z_n

By exciting the inputs of a device realizing the polynomial with the binary n-tuples thus determined, and inspecting the values at the output, we determine the values of the polynomial as shown in the column $P(x)$ in this table.

We first find at the zero-valued n-tuple the coefficient assigned to the first term of the polynomial, and then, by preliminary reading from the results for the unitary n-tuples the coefficients assigned to the variables, x_1, x_2 ... etc.

It follows that the test of the length $n+1$ is sufficient for testing a linear polynomial of n variables. The question is whether is possible to perform testing with fewer clocks.

We will use advantages of parallel calculations, which means simultaneous determination of several values of a function or a system of functions.

Parallel calculation of a function $f(x)=y$ is an operation

$$f\left(\mathop{*}_{i=1}^{k} x^i\right) = Y^i = \mathop{*}_{i=1}^{k} y^i$$

Similarly, for a system of m functions

$$P\left(\mathop{*}_{i=1}^{k} x^i\right) = \mathop{*}_{j=1}^{m}\mathop{*}_{i=1}^{k} y_j^i .$$

In this way, for a given system of three functions represented by the generalized disjunction, the values of the systems at four n-tuples of arguments $x^1=1$, $x^2=0$, $x^3=8$, $x^4=64$ can be determined by a single evaluation of the starting disjunction

$$F(x)=5\cdot 1\cdot 1\vee 3\cdot 8\vee 6\cdot 64 =5\vee 24\vee 384 = 413 = 110*011*101$$

This result represents the values of the following functions

$$f_3^3 f_2^3 f_1^3 * f_3^2 f_2^2 f_1^2 * f_3^1 f_2^1 f_1^1,$$

The result determined in this way implies that linear polynomials are fast realizable and easy testable.

Consider the application of arithmetic polynomials for representing of an important class of discrete devices, described by the model of switching automata (commutators).

Under the term of switching automata we understand a class of discrete devices (n,m) - commutators realizing a controlled transfer (commutation) of signals form n inputs to m outputs. We assume that switching automata involve all devices that can be realized by arbitrary linear polynomials.

In particular, we will consider the description of the function of commutators, since they comprise a significant share in the structure of multiprocessor systems.

By using arithmetic polynomials, it is possible to describe arbitrary logic functions (in general, systems of functions), and therefore, it is possible to describe the particular case of identical mappings realized by commutators. Real commutators perform as parallel devices in transmission of information. Their computer models are serial programs, and therefore the model of the function is slower than hardware realizations. It is understandable that a more successful mathematical description of the devices will accelerate the performance of their simulation.

Consider a number of devices related to commutators.

1. *Commutators realizing the functions of the form $y_j = f_j(x_i)$.*

A. Identical mapping of the cortege $X = \mathop{*}_{i=1}^{n} x_i$ into the cortege $Y = \mathop{*}_{j=1}^{n} y_j$ for $n = m$, $y_j = x_j$ is realized by the polynomial $Y = P(x) = \sum_{i=1}^{n} 2^{i-1} x_i$ (Fig. 1, *a*).

B. The same mapping, but with reversed order of functions $Y = P(x) = \sum_{i=1}^{n} 2^{n-i} x_i$ (Fig. 1, *b*).

Obviously, to the linear transformations indicated corresponds the operation of substitution.

C. Distribution of inputs. To the application of a variable to several inputs corresponds the addition of its orthogonal coefficients. For example, the following polynomial corresponds to the commutator in Fig. 1*c*, $Y = P(x) = (2^2 + 2^1)x_2 + x_1 = x_1 + 5x_2$.

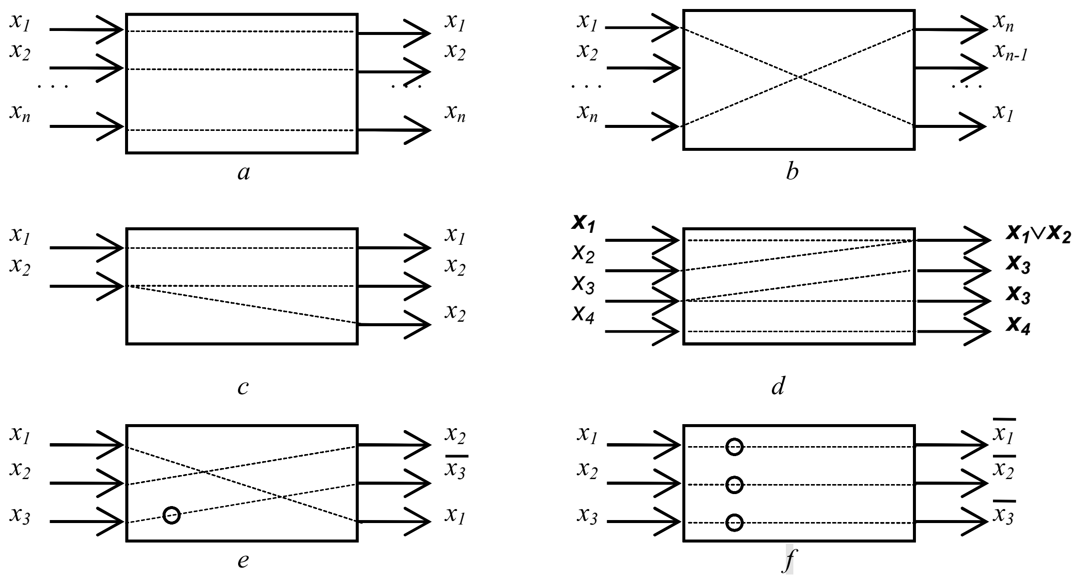

Fig. 1. Basic operations of commutators

D. Interconnecting of inputs, when transition of several variables to a single output corresponds to the disjunctions of interconnected variables (Fig. 1, *d*)

$$Y = x_4 * x_3 * x_3 * (x_1 \text{ v } x_2) = P(x) \\ = x_1 \text{ v } x_2 + 6x_3 + 8x_4 .$$

E. Transfer of the signal with inversion corresponds to the addition of 1 to signal (Fig. 1, *e*):

$$Y = x_1 * \bar{x}_3 * x_2 = 4x_1 + 2(1-x_3) + x_2 \\ = 2 + 4x_1 + x_2 - 2x_3 .$$

F. Inversion of all the input signals (Fig. 1, *f*) corresponds to the addition of the polynomial to the number 2^n-1

$$Y = \mathop{*}_{i=1}^{n} \bar{x}_1 = P(x) = 2^n - 1 - \sum_{i=1}^{n} 2^{i-1} x_i \ .$$

2. *Serial connection of commutators*

Let the commutator *A* realize a linear transformation of the cortege *X* into the cortege *Y*, and the commutator *B*, of the cortege *Y* into the cortege *Z*. Let us describe by means of the arithmetical polynomial the joint conversion, achieved by a series connection of commutators *A* and *B*.

We first fix two methods of the addition of terms in the linear polynomial.

A. Ordering by the input, when the terms are ordered according to the increasing order of their indices, for example, $2x_1 + 4x_2 + x_3$.

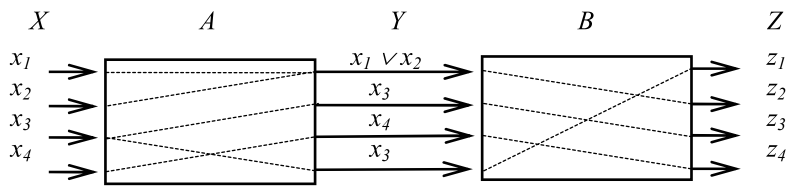

Fig. 2: Serial connection of commutators

B. Ordering by outputs, when the terms are ordered according to the increasing order of the coefficients, for example $x_3 + 2x_1 + 4x_2$.

It is clear that to the physical connection of commutators corresponds "identification" of outputs of the first as the inputs of the second. Therefore, to describe a serial composition of polynomials, written $P_A \circ P_B$, we should substitute the argument y_j (j = 1,...,n) in the polynomial $P_{B,}$ with the variable x_i, which by the polynomial P_A is described as the transition to the position *j*. Denote by *P* and *P′* the polynomials ordered by inputs and outputs, respectively. We order by output the polynomial realized by the commutator A:

$$P'_A = \sum_{i=1}^{n} a_i x_i$$

Let the other polynomial ordered by inputs take the form

$$P_B = \sum_{i=1}^{n} b_i y_i .$$

In this case, to the composition of polynomials $P_{AB} = P_A \circ P_B$ corresponds the polynomials with coefficients from the second polynomial and arguments from the first polynomial

$$P_{AB} = \sum_{i=1}^{n} a_i x_i \circ \sum_{i=1}^{n} b_i y_i = \sum_{i=1}^{n} b_i x_i .$$

Example. (See Fig. 2). Commutator A realizes the polynomial $P_A = x_1 \vee x_2 + 10x_3 + 4x_4$, and commutator B the polynomial $P_B = 2y_1 + 4y_2 + 8y_3 + y_4$.

We order by inputs the polynomial $P_A \to P_B = x_1 \vee x_2 + 2x_3 + 4x_4 + 8x_3$. For convenience, we write the polynomials under each other and perform the composition of inputs:

$P'_A = x_1 \vee x_2 + 2x_3 + 4x_4 + 8x_3$
$\circ$
$P_B = 2y_1 \quad + 4y_2 + 8y_3 + y_4$

$P_{AB} \ = 2(x_1 \vee x_2) + 4x_3 + 8x_4 + 1x_3$
$= 2(x_1 \vee x_2) + 5x_3 + 8x_4$

From this example, we define a generalization to the case of serial connection of k commutators.

Polynomial C, describing a k-level cascade, can be generated from coefficients of the polynomial B, describing the last commutator, and arguments of the polynomial A, ordered by outputs, describing the k-1-level cascade of the previous commutators. Therefore, addition of the third comutator W with the polynomial $P_w = \sum_{i=1}^{n} a_i z_i$ to the above described cascade, forms a composition of commutators (polynomials):

$P'_{AB} = x_3 \ + 2(x_1 \vee x_2) \ + 4x_3 \ + 8x_4$
$\circ$
$P_w = a_1 z_1 + a_2 z_2 + a_3 z_3 + a_4 z_3$

$P_{ABW} \ = a_1 x_3 + a_2 (x_1 \vee x_2) + a_3 x_3 + a_4 x_4$

From the above mentioned, it follows that linear logical polynomials have an interesting technical interpretation.

REFERENCE

[1] Mayugin, V.D., ”On polynomial realizations of corteges of Boolean functions”, *Doklady Russian Academy of Sciences*, Vol. 265, No. 6, 1338-1341.