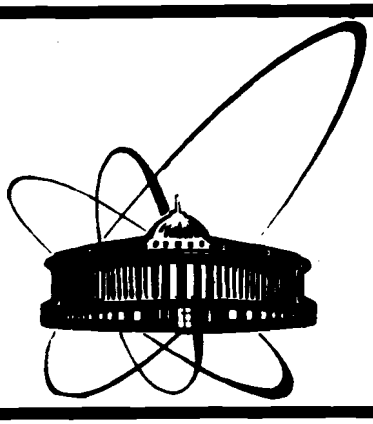


объединенный
институт
ядерных
исследований
дубна

E10-88-53

R.I.Gaidamaka, N.M.Nikityuk

APPLICATION OF ANALYTICAL TRANSFORMATIONS AND CALCULATIONS ON COMPUTER FOR SYNTHESIS OF SWITCHING FUNCTIONS AND SOLUTION OF THE PROBLEM OF DEVISING UNIVERSAL DYNAMICALLY PROGRAMMED LOGIC MODULES

Submitted to the International Conference ISSAC-88 and AAECC-6, Applied Algebra, Theory and Applications of Error Correcting Codes, Roma, July, 1988.

1988

1. THE CALCULATION OF SWITCHING FUNCTIONS REPRESENTED AS GALOIS FIELD GF (2^m) ELEMENTS ON COMPUTER

The theory of Galois field GF(2^m) being a natural continuation of the theory of Boolean field, the representation of switching functions as a polynomial, where both variables and coefficients are Galois field elements, is thought to be the most long-term one among the number of different methods of switching function synthesis.

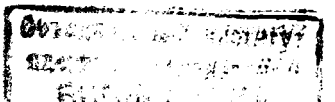
Representation of switching functions in such form (Galois switching function - GSF) has a number of advantages. First, algebraic operations can be executed over the GSF and the problem of minimization and its formal representation simplifies. Second, an input-output state of a combinational scheme or a sequentional automath being coded by the Galois field elements, a next state of input-outputs can be represented as a polynomial function of a current state and a current output. Third, the representation of switching function as a polynomial for a large number m(m · 3) makes it possible to use standard programming systems and present-day computers for calculation of logic schemes. Finally, the advantage of switching function represented as a polynomial is in compact form of presentation of multivalued and multilevel schemes.

For the purpose to approach the abstract Galois field theory to engineering practice all calculations of the schemes in each particular case have been executed and the possibility of descrete logic schemes designing on the base of analytical transformations and calculations on a computer are shown. The PL/I, REDUCE, SCOONSCHIP programs have been used[11]. The fundamental properties of GSF are considered in details in the literature[1-8].

Let us consider some examples. It is known that any switching function $f(x) = (x_0, x_1, x_2, \ldots, x_{m-1})$ of the m argument of the GF(2^m) can be presented as a polynomial[1-4]

$$f(x) = B(0) + A(1)x + A(2)x^2 + A(3)x^3 + \ldots + A(2^m - 1)^{2^m - 1}. \qquad (1)$$

Here and further a modulo-2 sum is denoted by the sign + and the A(k) coefficients are calculated from the expression



$$A(k) = \sum_{i=1}^{2^m-1} a_i^{-k} [B(0) + B(a_j)], \quad k = 1,2,3,\ldots,2^{m-1},$$

where $B(a_j)$ are the substitutional elements taken from the input-output correspondence Table and $B(0)$ is the function at a zero point. Thus for GSF synthesis the following steps are executed: an irreducible polynomial of the m /8/ power is chosen in conformity with a number of variables and all nonzero $GF(2^m)$ elements are found; a table of input-output correspondence is drawn up; the $A(k)$ coefficients are calculated; the $A(k)$ coefficients and the x powers are expanded in the basis elements of a selected field; similar terms are eliminated. Note that any x element of the Galois field $GF(2^m)$ can also be represented as a polynomial

$$x = x_0 a^0 + x_1 a^1 + x_2 a^2 + \ldots + x_{m-1} a^{m-1},$$

where $a^0, a^1, a^2, \ldots, a^{m-1}$ are basis elements of the field and $x_0, x_1, x_2, \ldots, x_{m-1}$ are equal to 1 or 0.

Example 1. Consider a Galois field $GF(2^3)$ formed over the irreducible polynomial $x^3 + x + 1$. Suppose that $a^0 = 100$, $a^1 = 010$, $a^2 = 001$ are basis elements (linear independing) of the field, and a^1, a root of the polynomial. In such a way one can easy find the other field elements because $a^4 = a^3 a^1 = a^2 + a^1 = 011$, $a^5 = a^4 a^1 = a^3 + a^2 = a^2 + a^1 + a^0 + a^2 = 111$, $a^6 = a^5 a^1 = a^2 + a^0 = 101$ and $a^7 = a^0$.

Suppose that a scheme of a one-bit full summator should be designed. Let's draw up a table of correspondences (Table 1). The sequence of the field elements arranged in increasing order of their powers and their binary equivalents (inputs) are given on the left. The corresponding values which it is necessary to get at the outputs of the summator are shown in the second column.

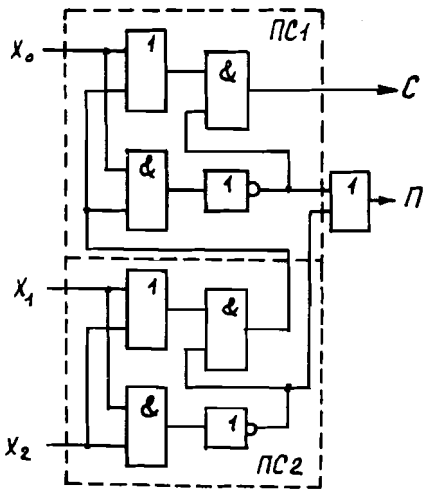


Fig. 1. Principle scheme of 1-bit combined summator ПС1, ПС2 *- halfsummator.*

Table 1.

Inputs	Outputs
$x = x_0, x_1, x_2$	$B(a_j)$
$0 = 000$	$0 = 000 = 00$
$a^0 = 100$	$a^0 = 100 = 10$
$a^1 = 010$	$a^0 = 100 = 10$
$a^2 = 001$	$a^0 = 100 = 10$
$a^3 = 110$	$a^1 = 010 = 01$
$a^4 = 011$	$a^1 = 010 = 01$
$a^5 = 111$	$a^3 = 110 = 11$
$a^6 = 101$	$a^1 = 010 = 01$
$a^7 = 100 = a^0$	$a^0 = 100 = 10$
	Carry Sum

x_0, x_1, x_2 are values of the first summand and the second one and the carry output, C and 11 are values of the sum and the carry at the outputs of the summator. For our example, as follows from Table 1, the substitutional elements B(1), B(2), B(3), B(4), B(5), B(6), B(7) are the Galois field elements $a^0, a^1, a^2, a^3, \ldots, a^6$, accordingly. This is a more detailed presentation of the calculation of the A(1) coefficient:

$$A(1) = \frac{a^0}{a^0} + \frac{a^0}{a^1} + \frac{a^0}{a^2} + \frac{a^1}{a^3} + \frac{a^1}{a^4} + \frac{a^3}{a^5} + \frac{a^1}{a^6} = a^0 + a^0 a^6 + a^0 a^5 + a^1 a^4 +$$

$$+ a^1 a^3 + a^3 a^2 + a^1 a^1 = a^0.$$

Here the division operation of two elements was being exchanged for the multiplication by an inverse element according to their rules in the Galois field theory. Then we have

$$A(2) = \frac{a^0}{(a^0)^2} + \frac{a^0}{(a^1)^2} + \frac{a^0}{(a^2)^2} + \frac{a^1}{(a^3)^2} + \frac{a^1}{(a^4)^2} + \frac{a^3}{(a^5)^2} + \frac{a^1}{(a^6)^2} = a^1 = 010.$$

The analogous calculations give $A(3) = a^0 = 100$, $A(4) = A(7) = 0$, $A(5) = a^4 = 011$, $A(6) = a^6 = 101$. By virtue of these calculations the expression (1) has the form

$$f(x) = x^1 + ax^2 + x^3 + a^4 x^5 + a^6 x^6. \qquad (2)$$

The expression (2) can be simplified by means of representation of both the coefficients and the variables as a polynomial in the basis elements. For example $a^4 = a^1 + a^2$, $a^6 = a^0 + a^2$, $x_0^2 = x_0 + x_2 a^1 + (x_1 + x_2)a^2$ and so on. Hence, we have

$$f(x) = (x_0 + x_1 a^1 + x_2 a^2) + a^1[x_0 + x_2 a^1 + (x_1 + x_2)a^2] + (x_0 + x_1 + x_2 + x_1 x_2) +$$

$$+[(x_1 + x_0 x_1 + x_0 x_2)a^1 + (x_2 + x_0 x_1)a^2] + (a^1 + a^2)[(x_0 + x_1 + x_2 + x_1 x_2) +$$

$$+(x_1 + x_2 + x_0 x_2)a^1 + (x_1 + x_0 x_1 + x_0 x_2)a^2 + (a^0 + a^2)[(x_0 + x_1 + x_2 + x_1 x_2) +$$

$$+(x_2 + x_0 x_1)a^1 + (x_1 + x_2 + x_0 x_2)a^2].$$

By simple manipulations we get the following bool expressions. By means of these expressions the work of the one-bit full summator is described:

$C = x_0 + x_1 + x_2 \quad \langle a^0 \rangle$

$II = x_0 x_1 + x_0 x_2 + x_1 x_2 \quad \langle a^1 \rangle$.

It is shown by practice for $m > 4$ a large body of calculations is increasing. That is why a computer should be used in this case.

Example 2. Let us calculate the scheme of the sequential automath. The Galois field $GF(2^4)$ elements generated over the irreducible polynomial $x^4 + x + 1$ arrange correspondingly in increasing order of their powers at the inputs. At the outputs we obtain the same elements in the given sequence (as shown in (Table 2). The Galois field $GF(2^4)$ elements in increasing order of their powers can be rather simply obtained with the help of the counter in the $GF(2^4)$. It is a shift register with the logical opposite connections. If we carry unit into the low-order digit and zeros into the other ones the successive shifts of the register will give us the presentation of the a^1 element powers and the root of the polynomial $x^4 + x^1 + 1$ as they are shown in Table 2 on the left. It should be noted that any x element of the Galois field $Gf(2^4)$ has the form $x_0 a^0 + x_1 a^1 + x_2 a^2 + x_3 a^3$. To construct a scheme for transformations of 4-bit codes in accordance with Table 2 it is essential to calculate 15 coefficients in the polynomial GSF representation of 4 variables:

Table 2

Inputs	Outputs
$x = x_0, x_1, x_2, x_3$	$f(x)$
$0 = 0000$	0
$a^0 = 1000$	$a^1 = 0100$
$a^1 = 0100$	$0 = 0000$
$a^2 = 0010$	$a^7 = 1101$
$a^3 = 0001$	$a^5 = 0110$
$a^4 = 1100$	$a^5 = 0110$
$a^5 = 0110$	$a^{11} = 0111$
$a^6 = 0011$	$a^{13} = 1011$
$a^7 = 1101$	$a^0 = 1000$
$a^8 = 1010$	$a^3 = 0001$
$a^9 = 0101$	$a^{14} = 1001$
$a^{10} = 1110$	$a^{14} = 1001$
$a^{11} = 0111$	$0 = 0000$
$a^{12} = 1111$	$a^2 = 0010$
$a^{13} = 1011$	$a^4 = 1100$
$a^{14} = 1001$	$a^0 = 1000$

$$f(x_0, x_1, x_2, x_3) = B(0) + A(1)x + A(2)x^2 + A(3)x^3 + A(4)x^4 + A(5)x^5 +$$

$$+ A(6)x^6 + A(7)x^7 + A(8)x^8 + A(9)x^9 + A(10)x^{10} + A(11)x^{11} + A(12)x^{12} + (3)$$

$$+ A(13)x^{13} + A(14)x^{14} + A(15)x^{15}.$$

By calculation of the A(k) coefficients and elimination of similar terms on the computer we get the following switching functions. With the aid of these functions a scheme of sequential automath shown in Fig.2 has been obtained. Such schemes can be used to get a given sequence of binary digits for example in microprogramming control devices

$$x_1 + x_0 x_2 + x_0 x_3 + x_1 x_2 + x_1 x_3 + x_0 x_1 x_3 + x_1 x_2 x_3 + x_0 x_1 x_2 x_3 \quad \langle a^0 \rangle$$

$$x_0 + x_2 + x_3 + x_1 x_3 + x_0 x_1 x_3 + x_1 x_2 x_3 \quad \langle a^1 \rangle \quad (4)$$

$$x_3 + x_0 x_1 + x_0 x_3 + x_1 x_3 + x_1 x_2 x_3 + x_0 x_1 x_2 x_3 + x_1 x_2 \quad \langle a^2 \rangle$$

$$x_2 + x_1 x_3 + x_0 x_1 x_3 + x_0 x_2 x_3 \quad \langle a^3 \rangle$$

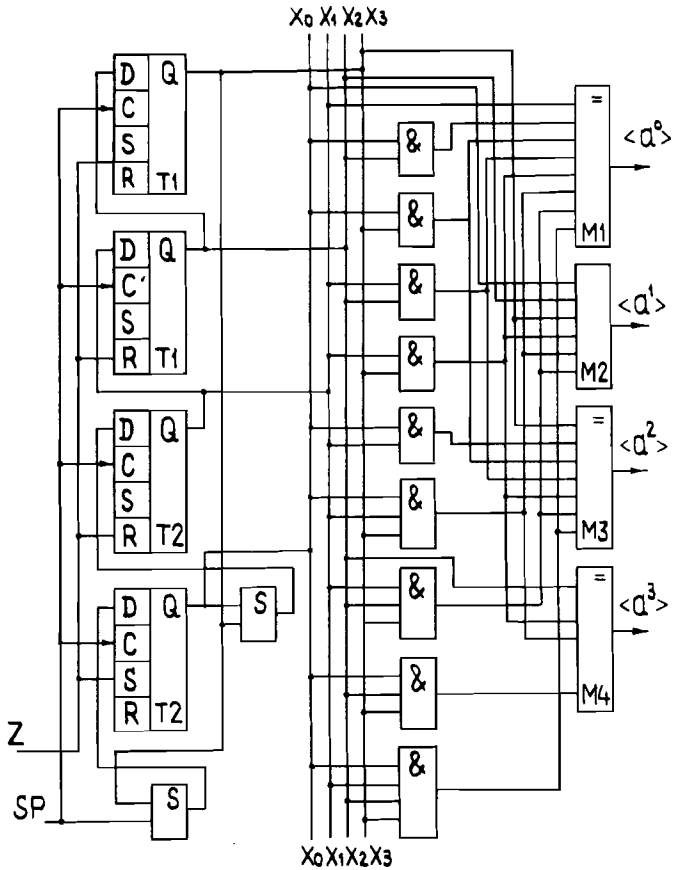


Fig.2. Principle scheme of sequential automath executing equation (4). T1, T2, are D-triggers, S is modulo 2 summator, M1÷M4 are SN74180 microcircuits (parity check circuits).

For simultaneous modulo-2 summation of several summands, a SN74180 microcircuit (Fig.3) having 8 data inputs or a MC100160 microcircuit having 12 inputs (Parity checker) can be used. 64 and 144 one-bit summands respectively can be summarized simultineously with 2-cascade turning on of a parity checker. All necessary calculations for example 2 have been executed by means of the ES-1033 computer. For this purpose the PL-1 program which fulfils calculations in Galois field are given. At the present stage the program has such possibilities that the total number of symbols in the factors must not

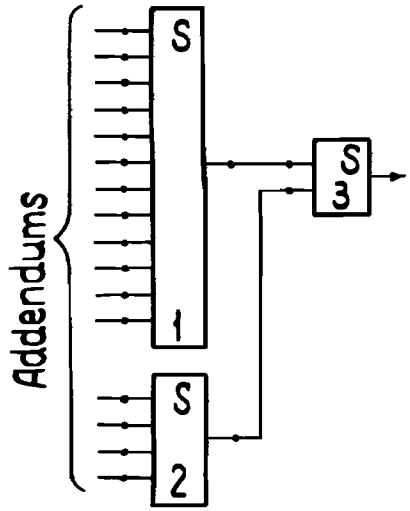


Fig.3. Principle scheme of modulo 2 summator for 16 inputs, 1, 2 are MC10160 microcircuits; 3, MC10107 microcircuits.

be in excess of 250 during the calculation of coefficients for every given basis. For m = 4 and m = 5 a processor spends 1 minute and 5 ones accordingly.

Example 3. The calculation of a scheme for a sum of two 3-bit digits with a carry. It is obvious that such scheme has 6 inputs for summands (an input for a carry from the low-order digit is thought to be absent), 2 outputs for sum and an output for carry. It means that the calculations must be executed over switching function of 6 variables in the Galois field (2^6) which consists of 63 nonzero elements. These elements are considered a 6-bit binary digits or as arguments of the function of 6 variables x_0, x_1, x_2, x_3, x_4, x_5.

Choose a polynomial of the 6 power $x^6 + x + 1$. The tables of irreducible polynomials to the 34 power are given in [8]. With some calculations and simplification on computer the following expressions are obtained

$$x_0x_3 + x_0x_1x_1 + x_1x_3x_4 + x_0x_1x_2x_5 + x_0x_2x_4x_5 + x_1x_2x_3x_5 + x_2x_3x_4x_5 \qquad <a^0>$$

$$x_0 + x_3 + x_0x_4 + x_1x_2x_5 + x_2x_4x_5 \qquad <a^1> \qquad (5)$$

$$x_1 + x_4 + x_2x_5 \qquad <a^2>$$

$$x_2 + x_5 \qquad <a^3>$$

Fig.4 presents a principle scheme of a 3-bit parallel summator and a mod-2 summator. With the aid of the first one $x_0x_1x_2$ and $x_3x_4x_5$ are summed up in the "and" basis.

It should be noted that for a large number of variables the draving-up of input-output correspondence Table can be automated.

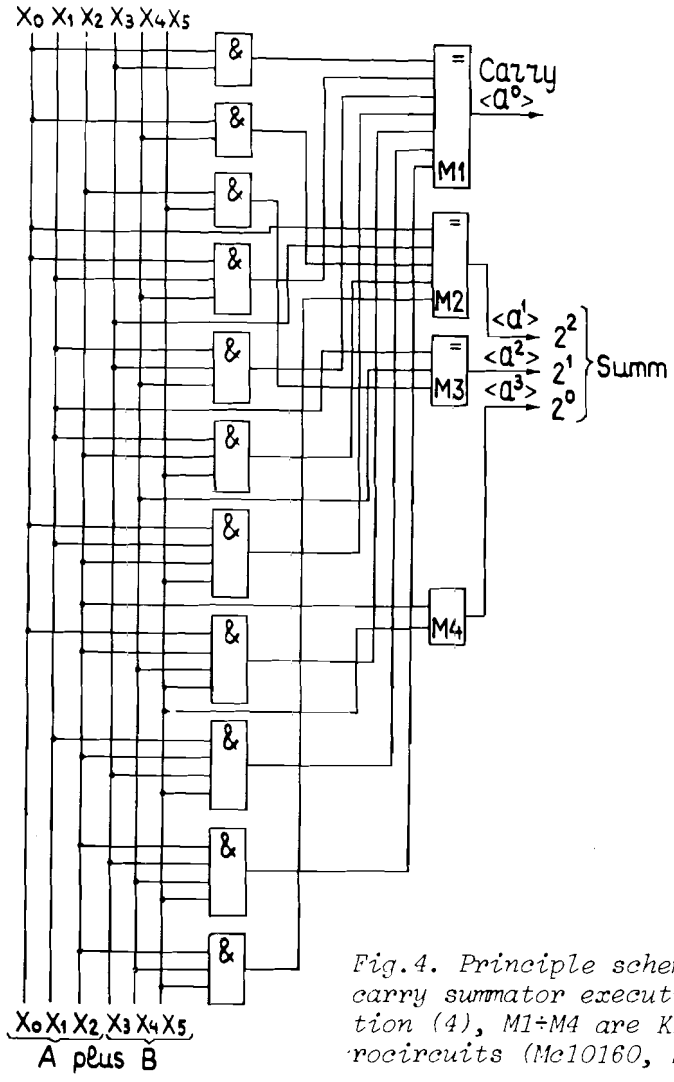


Fig.4. Principle scheme of 3-bit carry summator executing equation (4), M1÷M4 are K155IP2 microcircuits (Mc10160, SN74180).

2. UNIVERSAL DYMANICALLY PROGRAMMABLE MODULE

Nowadays intensive work that is of interest from both theoretical and practical points of view is carried out with the object of devising universal dynamically programmble modules (UDPLM) [14, 24]. The main problem is to design an efficient mathematical device. It allows to create the universal modules and the specialized modules as well. Then using minimum quantity of tuning inputs we must design a module that executes a highest possible number of logic operations with a smallest number of logic elements to be used. Below we shall consider in detail that analytical calculations on computer can be used with that end in view.

The identity (3) gives the following:

the values of the A(1)-A(15) coefficients determine a type of a GSF to be realized;

to design a scheme for calculation of the polynomial in the (1) identity a device to multiply, to sum up and to raise to power Galois field elements should be created. As shown in the investigation [14] such operations can be executed with the standard microcircuits of medium-scale integration and large-scale integration as well. To execute these operations a table can be draw up. Fig.5 presents a table for multiplication of the A and B elements in the Galois field GF(2^4). The field elements for a fixed m forming a cycle group, calculations of more difficult expressions such as simultaneous multiplication of several factors and simultaneous multiplication with rai-

A →

B ↓ x	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}
a^0	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}
a^1	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0
a^2	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1
a^3	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2
a^4	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3
a^5	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4
a^6	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5
a^7	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6
a^8	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7
a^9	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8
a^{10}	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9
a^{11}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}
a^{12}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}
a^{13}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}
a^{14}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}

Fig.5. Table for multiplication of two elements in the GF(2^4) modulo $x^4 + x + 1$.

B \ A →	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}
a^0	a^0	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}
a^1	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}
a^2	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}
a^3	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0
a^4	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1
a^5	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2
a^6	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3
a^7	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4
a^8	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5
a^9	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6
a^{10}	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7
a^{11}	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8
a^{12}	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9	a^{12}	a^0	a^3	a^6	a^9
a^{13}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}	a^{13}	a^1	a^4	a^7	a^{10}
a^{14}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}	a^{14}	a^2	a^5	a^8	a^{11}

Fig.6. Table for multiplication of the B element by A^3 in the GF(2^4) modulo $x^4 + x + 1$.

sing to power attached can be reduced to table methods. Table of all the possible values of the A B^3 expression is given in Fig.6. The upper row consists of the A elements, the next row consists of those to the power. The B elements (the A, B elements are the Galois field GF(2^4) ones, see Table 2) are given in the left column. For example the B A^3 expression for $A = a^7$ and $B = a^{12}$ is equal to the a^3 element. Indeed, $a^{12} \times (a^7)^3 = a^{12} a^{21} = (a^{33}) = a^{15} a^{15} a^3 = a^3$.

In this way 14 tables of the same type can be drawn up.

Figure 7 presents the UDPLM scheme that has been designed accordingly to the identity (3). The scheme contains 4 input variables, 15 inputs for tuning coefficients and 4 output[16]. To apply this device 15 schemes for simultaneous multiplication and raising to power of two Galois field GF(2^4) elements should be available. If we use a fast PROM MC10149 with that end in view, we obtain the identical and rather simple scheme of this type. The difference is in the content of PROM. Figure 8 presents such a scheme. It consists of a 4-bit register that contains the tuning coefficients and the MC10149 micro-

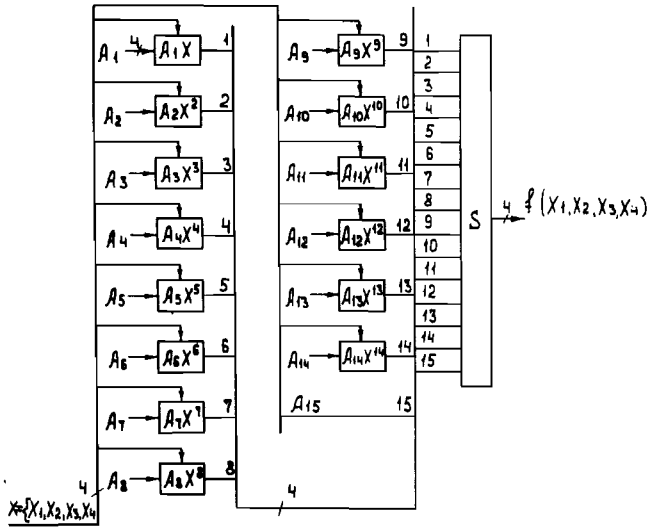


Fig.7. UNPLM scheme for a function of 4 variables, S is modulo 2 summator.

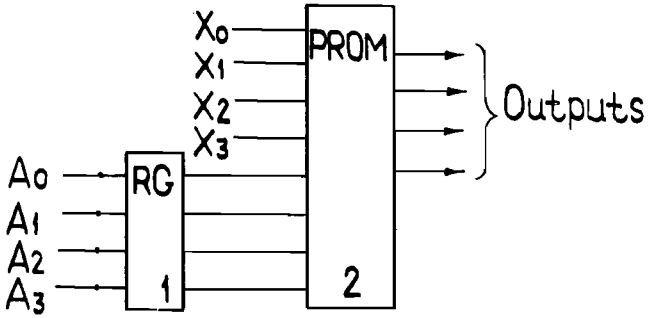


Fig.8. Scheme of reasing to power and simultaneous multiplication of an element. 1 is a register; 2, PROM MC10149.

circuits. Changing the A(k) coefficients, where k =1,2,3,...,15, by hand or by computer the module can fast be reconstructed to perform different functions up to 65536 in number.

Consider some examples which are illustrated by Table 3.

In the first column on the left the Galois field GF(2^4) elements and their binary equivalents are given. The signals corresponding to their codes are driven to 4 inputs. The values corresponding to them at the outputs and the A(k) coefficients calculated on the computer are shown in the next co-

Table 3. Representation of logic functions by the Galois field GF(2^4) elements

Inputs/function	Coincidence		Pass		Inversion	
$x = x_0, x_1, x_2, x_3$	$f(x)=x_0 x_1 x_2 x_3$	A(k)	f(x)=x	A(k)	$f(x)=x_0 x_1 x_2 x_3$	A(k)
a^1 = 0100	0000	a^3	a^1	a^o	1011	a^o
a^2 = 0010	0000	a^6	a^2	0	1101	0
a^3 = 0001	0000	a^9	a^3	0	1110	0
a^4 = 1100	0000	a^{12}	a^4	0	0011	0
a^5 = 0110	0000	a^o	a^5	0	1001	0
a^6 = 0011	0000	a^3	a^6	0	1100	0
a^7 = 1101	0000	a^6	a^7	0	0010	0
a^8 = 1010	0000	a^9	a^8	0	0101	0
a^9 = 0101	0000	a^{12}	a^9	0	1010	0
a^{10} = 1110	0000	a^o	a^{10}	0	0001	0
a^{11} = 0111	0000	a^3	a^{11}	0	1000	0
a^{12} = 1111	1000	a^6	a^{12}	0	0000	0
a^{13} = 1011	0000	a^9	a^{13}	0	0100	0
a^{14} = 1001	0000	a^{12}	a^{14}	0	0110	0
$a^{15} = a^o$ =1000	0000	a^o	a^o	0	0000	a^{12}

lumns. The methods of calculation of the coefficients are given in [9]. The function is supposed to have a true value at the outputs if it is equal to a unity element a^o = 1000. In the first case the coefficients calculated on the computer equal to A(1) = a^3, A(2) = a^6, A(3) = a^9..., A(14)= a^{12}, A(15)= = a^o. Substituting into expression (3) we obtain

$$f(x) = a^3 x + a^6 x^2 + a^9 x^3 + a^{12} x^4 + a^o x^5 + a^3 x^6 + a^6 x^7 + a^9 x^8 + a^{12} x^9 + a^o x^{10} + a^3 x^{11} + a^6 x^{12} + a^9 x^{13} + a^{12} x^{14} + a^o x^{15} . \quad (6)$$

The fact that work of a 4 inputs scheme of coincidence is described by means of expression (6) can be verified in two ways:
1. Substituting $x = a^{12}$ and calculating $a^3 a^{12} = a^{15} a^o$, $a^6 (a^{12})^2 = a^{30} = a^{15} a^{15} = a^o$ and so on we obtain 15 terms each of those being equal to a^o. Taking a sum of these terms to modulo 2 we get finally $f(x) = a^o$. Substituting the other elements into expression (6) instead of x we obtain 15 different elements $a^o \div a^{14}$. The sum of all the field GF(2^4) elements to modulo 2 equals to zero.
2. Let us simplify expression (6) expanding the field elements of different degress in the basis elements by the computer. As a result of the simplification one can see the next expression on terminal:

$$f(x) = (x_1 x_2 x_3 x_4) = a^o .$$

It means that the module outputs equal to a^o. It is a true value if all the inputs are in a state of logic unity.

Consider the second column. A module tuning in performance of a "passive" operation when the input-output values are the same (repeating logic signals) is illustrated in it. In this case only coefficient for x is equal to a^o. We have

$$f(x) = a^o x = a^o (a^o x_o + a^1 x_1 + a^2 x_2 + a^3 x_3) = x .$$

Substituting the corresponding coefficients a^o for x and a^{12} for x^{15} in identity (3) we shall get a logic equation for inverse with the exception of a zero point:

$$f(x) = a^o x + a^{12} x^{15} = x + a^{12} .$$

CONCLUSION

The calculations of switching functions represented here are generally experimental ones. The data obtained prove the truth of usage of present-day computers for the automatical synthesis of switching functions for a large number of variables $m(2 \leq m < 12)$. The main improvement should be directed to descrease both computing time of central processor and required size of memory. There are large reserves in that way. The operations of raising the elements to power and expanding them in basis elements to simplify the expressions are most laborious ones. Algebra in Galois field $GF(2^m)$ being modular one, it is in a position to perform them by selected irreducible polynoms beforehand and to store results in external memory.

It can be seen in our examples that using this method of a synthesis of switching functions "and" elements, modulo-2 summators and many-input parity check circuits which consist of the modulo-2 summators of the same type for 2 inputs are used as the basis elements.

A microcircuit technique extending, we pay a great attention to the problem of a synthesis of circuits in which "and" elements and modulo-2 summators are used [19, 20]. In particular the following facts account for it:

a component of inversion containing within summator, it is by no means to introduce an inversion of variables;

canonical representation of Reed-Maller switching functions can also be executed more simply within basis "and" exepting "or" one;

the logic schemes which realize switching functions within the "and" basis and modulo-2 summator are obtained more simply compared with the "and", "not" (as a practice shows) [19] but the question is still "opened" theoretically. This fact is also important for devising multifunctional LSI which consist of modulo-2 summators;

analogous basis is widely used for devising universal generator of functions and dynamically programmable modules as well [24].

The calculation of Galois switching functions by computer enables for integrated automation of engineering systems of discrete logics. A new method of devising UDPLM is given. Its scheme has an algebraic structure.

REFERENCES

1. Gill A., Jacob Y.P. Quarterly of Applied Mathematics, 1966, v.XXXIV, April No.1, p.57.
2. Benjauthrit B., Reed I. - TEEE Trans. on Computers, 1978, v.C-27, No.8, p.757.
3. Menger K.C. - TEEE Trans. on Computers, 1969, v.C-18, No.3, p.241-250.
4. English W.R. - IEEE Trans. on Computers, 1981, v.C-30, No.3, p.225-229.
5. Bartee T.C., Schneider P.I. - Information and Control, 1963, v.6, No.1, p.79-98.
6. Tanaka H., Kasahara M., Tezuka Y., Kasahara Y. - Information and Control, 1968, v.13, No.1, p.75-84.
7. Nikityuk N.M. JINR P11-80-484, Dubna, 1980.
8. Peterson W. Error-Correcting Codes, "Mir", Moscow, 1964.
9. Alexandrov I.N., Gaidamaka R.I., Nikityuk N.M., Shirikov V.P. JINR P10-86-365, Dubna, 1986.
10. Gaidamaka R.I., Nikityuk N.M., Shirikov V.P. - In: Analytical Calculations on Computers and Their Application in Theoretical Physics. JINR D11-83-511, Dubna, 1983, p.246-252.
11. Gaidamaka R.I., Nikityuk N.M., Shirikov V.P. JINR P10-84-841, Dubna, 1984.
12. Alexandrov I.N., Gaidamaka R.I., Nikityuk N.M. - In: Analytical Calculations on Computers and Their Application in Theoretical Physics. JINR D11-85-791, Dubna, 1985, p.295.
13. Nikityuk N.M. JINR P11-87-54, Dubna, 1987.
14. Nikityuk N.M. Avt. svid. No.1236457 (USSR), in OI, 1986 No.21, p.199 (in Russian).
15. Suares R.E., Chang O., Adam V. - IEEE Transaction on Computers, 1981, v.C-30, No.1, p.79.
16. Hurst S.L. - IEEE Transaction on Computers, 1981, v.C-30, No.12, p.986.
17. Nikityuk N.M. Avt.svid. No.1234861 (USSR), in OI, 1986, No.20, p.229 (in Russian).
18. Nikityuk N.M. JINR P11-85-365, Dubna, 1985, p.10.
19. Mikhopadhyay A., Schmitz G.- IEEE Trans. on Computers, 1970, v.C-19, No.2, p.132-140.
20. Schmookler M.S. - IEEE Trans. on Computers, 1969, v.C-18, No.10, p.957.
21. Swamy S. - IEEE Trans. on Computers, 1972, v.C-20, No.9, p.1008-1009.
22. Kodanpani K.L. - Electronic Letters, 1973, v.9, No.13, June 1973, p.296-287.

23. Pradhan D., Patel M. - IEEE Trans. on Computers, 1975, v.C-24, No.2, p.206-210.
24. Suarez R.E., Chang O., Adam V. - IEEE Trans. on Computers, 1981, v.C-30, No.1, p.79-81.

Received by Publishing Department
on January 21, 1988.

Гайдамака Р.И., Никитюк Н.М. E10-88-53

Применение аналитических преобразований и расчетов на ЭВМ для синтеза переключательных функций и решения проблемы создания универсальных динамически программируемых логических модулей

В данном докладе рассмотрены некоторые вопросы расчета переключательных функций при помощи ЭВМ. При этом входные и выходные параметры переключательной функции представлены в виде элементов поля Галуа $GF(2^m)$ и это позволяет любую функцию из m аргументов представить в виде полинома степени 2^m-1. Рассмотрено несколько примеров расчета конкретных схем с помощью аналитических вычислений на ЭВМ.

Работа выполнена в Лаборатории высоких энергий ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1988

Gaidamaka R.I., Nikityuk N.M. E10-88-53

Application of Analytical Transformations and Calculations on Computer for Synthesis of Switching Functions and Solution of the Problem of Devising Universal Dynamically Programmed Logic Modules

The questions of calculation of switching functions by means of computers are considered. The input and output variables are elements of the Galois field $GF(2^m)$ that allows to present any switching function as a polynomial to power 2^m-1. The examples of the calculation of the schemes in each particular case are presented.

The investigation has been performed at the Laboratory of High Energies, JINR.

Preprint of the Joint Institute for Nuclear Research. Dubna 1988