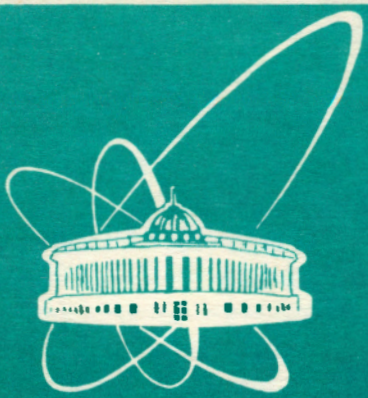

объединенный
институт
ядерных
исследований
дубна

E10-93-412

I.N.Aleksandrov, V.M.Kotov, N.M.Nikityuk

SOME QUESTIONS OF AN APPLICATION OF GALOIS FIELDS GF(2^m) SWITCHING FUNCTIONS

Submitted to «Автоматика и телемеханика»

1993

1 INTRODUCTION

Recently polynomial forms of the representation of Boolean functions in the form of the Zhegalkin polynom have found wider application to the problems of analysis and synthesis of them by means of truth-table combinatorial layouts [1],[2]. These forms have a homogeneous algebraic structure and are well realized in modern microelectronic. However, capacity of calculating the coefficients in the Zhegalkin polynoms is high and grows essentially with increasing number of variables. In [3],[4] it is shown that the system of Boolean functions is made to represent in the form of generalizing arithmetic polynom, which allows the parallel calculation of Boolean functions system more conveniently. The method of constructing polynoms in Galois field GF(2^m) is also known, which is based on the interpretation of inputs and outputs of switching layout as field elements. This direction is investigated in [5]-[7]. In [5] an expression for immediate calculation of polynomial decomposition coefficients is given. The results of this work were used for the creation of switching layouts with a goal of a combinatorial summator and sequential automaton synthesis [8]. The goal of the work is to show the efficiency of polynomial decomposition in the Galois fields for the synthesis of UDPLM .

2 Basic attributes and definitions. Basic decomposition theorem

The Galois fields are the natural extention of the Boolean field. They were well investigated and have a wide spectrum of applications [9]-[11]. Any switching function with m inputs and m outputs has no more than 2^m values. Therefore, it is given over finite field, frequently called Galois field GF(p^m). The number p is called a field basis and must be simple. We shall consider the case $p = 2$, however all results are valid for other simple p, therefore the actuality of the given direction is increasing with the creation of maltivalued logic devices. Let us introduce a field of coefficients GF(2) with elements 0 and 1, modulo 2 addition as a field's addition operation and a conjuction as a field's multiplication operation. In this field the operations of addition and subtraction are completely identical.

Introduce over GF(2) a field GF(2^m) as a field of polynoms with coefficients from GF(2) and degree less than m. The primitive root of irreducible polynom of degree m with its all degrees will be play the role of variables. There is a belief in this case that the irreducible polynom generates a field. The root is called primitive, if within a set of all its 2^m different degrees they do not coincide. Thus, the degrees of the primitive polynom cover all the field GF(2^m). The addition operation in this field is the usual operation of polynom's addition, where the coefficients are added in the field GF(2). Multiplication is a multiplication of polynoms in modulo of a bearing polynom. In this field all the usual field's axioms are valid. Let us give additionally the attributes of the finite fields which will be useful for understanding the paper (in more detail see [5]).

Attribute 1. For any $X \in GF(2^m)$: $X + X = 0$.

Attribute 2. For any nonzero $X \in GF(2^m)$: $X^{m-1} = 1$.

Attribute 3. For any $X, Y \in GF(2^m)$: $(X+Y)^2 = X^2 + Y^2$.

Attribute 4. For any non-one $X \in GF(2^m)$: $\sum_{k=1}^{2^m-1} X^k = 1$.

The first m degrees of primitive root α^0,α,α^2, α^3,...,α^{n-1} are linearly independent and can be given as a basis of the Galois field. This means that any field 's element X can be represented in the form:

$$X = a_0 * \alpha^0 + a_1 * \alpha + \cdots + a_{m-1} * \alpha^{m-1}, \tag{1}$$

where $a_i \in 0, 1$. If we take any binary number $(a_0, a_1, \ldots, a_{m-1})$ of length m as a set of coefficients a_i from (1), we may interpret it as the element of the Galois field.

THEOREM 1(expansion). Any switching function f(X) can be represented in the form (and in a unique way)

$$f(X) = f(0) + \sum_{i=0}^{2^m-1} G_i X^i, \tag{2}$$

$$G_i = \sum_{j=0}^{2^m-1} \alpha_j^{-i} f_j, \tag{3}$$

where $f_j = f(\alpha_j)$, and $\alpha_j = \alpha^j$ - j-th degree of α.

The proof of the theorem is given in [5]. In [6] theorems of the same type for the case of multivalued functions are given. Theorem 1 is a basis for the creation of UDPLM.

3 The creation of polynom for dynamically adjusted switching layout

Starting from (1), we can calculate in the general form:

$$\begin{aligned} X &= a_0\alpha^0 + \cdots + a_{m-1}\alpha^{m-1}, \\ X^2 &= (a_0\alpha^0 + \cdots + a_{m-1}\alpha^{m-1})(a_0\alpha^0 + \cdots + a_{m-1}\alpha^{m-1}), \\ &\vdots \\ X^{2^m-1} &= (a_0\alpha^0 + \cdots + a_{m-1}\alpha^{m-1})^{2^m-1}. \end{aligned}$$

When performing raising to a power by means of multiplication of (1) by itself (appropriate number of times), one gets all the X^i from (2) in polynomial form. If for a concrete switching function coefficients G_i are calculated from formula (3) and its values with degrees of X in polynomial form are substituted into (2), after eliminating of equal values we can get a Zhegalkin polynom system. Each polynom will be represented as a coefficient of basic element α^i. This method of getting polynoms is described in [8],[12]. However, in this case the values of G_i will be hardwired in layout, the general form will be a little simpler but the possibility of adjusting the layout for different functions will be lost. For UDPLM together with variable X, coefficients G_i should be inputs for the module.

Let us represent G_i in the general polynomial form through basis as we have represented X in (1):$G_i = b_{i_0}\alpha^0 + \cdots + b_{i_{m-1}}\alpha^{m-1}$. The given expressions with the expressions for all X^i are substituted into (2). As a result we will get the required polynom for UDPLM, in which X and all G_i are variables given via expansion in the basis. On its basis one can synthesize UDPLM, X and all G_i being inputs for layout. To adjust UDPLM to a concrete function it is enough to calculate values of all G_i for this function from formula (3) and to load them into storage registers. The values of G_i are directed to the inputs of layout together with the values of X during operating UDPLM . In appendexes 1,2 the polynomial form of X^k and GX^k accordingly is given for m=4 and the irreducible polynom $X^4 = X + 1$ (table of the irreducible polynoms for $m \leq 34$ is given in [9]). The expressions obtained for X^k are bulky, therefore all the calculations should be performed on computer, which has been done by the authors.

Using the fact that $X^k = XX^{k-1}$, the layout of UDPLM can be simplified by means of increasing the number of cascades in it, i.e. by

increasing the delay time. For example, only even parity degrees of X can be realized (their expressions are in average simpler), and odd parity degrees are obtained on the further cascade by means of multiplication $X^{2l+1} = YX$, where Y is outputs from layout of X^{2l} and l=1,2,...,$(2^{m-1} - 1)/2$. Different methods of the realization of multiplication, division and addition operations are given in [13]-[19].

Using the fact that expressions for X^{2^l}, l=0,1,...,m-2 are essentially easier (see appendix 1, it is the result of attribute 3), one can also simplify the layout by means of increasing the number of cascades.

EXAMPLE 1. For the Galois field GF(2^4) and the irreducible polynom $x^4 = x + 1$ formula (2) has the following form:

$$F(X) = F(0) + G_1X + G_2X^2 + \cdots + G_15X^15 = F(0) + [G_1X + G_2X^2 + G_3X^3] +$$
$$X^4[G_4 + G_5X + G_6X^2 + G_7X^3] + X^8[G_8 + G_9X + G_{10}X^2 + G_{11}X^3] +$$
$$X_{12}[G_{12} + G_{13}X + G_{14}X^2 + G_{15}X^3].$$

Substituting the expressions for G_i and X^k from appendixes 1,2, we get a two-cascade layout of the calculation of any 4-input switching function. The first cascade represents the calculation of the expressions in brackets, the second is the realization of the remaining operations of multiplication and addition. However, the expressions for X^3, X^{12} are still large. Using only the expressions for X, X^2, X^4, X^8, we get the following expressions for F(X):

$$F(X) = F(0) + G_1X + G_2X^2 + G_3X^2X + G_4X^4 + G_5X^4X + G_6X^4X^2 +$$
$$G_7X^4X^2X + G_8X^8 + G_9X^8X + G_{10}X^8X^2 + G_{11}X^8X^2X + G_{12}X^8X^4 +$$
$$G_{13}X^8X^4X + G_{14}X^8X^4X^2 + G_{15}X^8X^4X^2X =$$
$$F(0) + [X^8(G_8 + G_{12}X^4)] + [G_1 + X^8(G_9 + G_{13}X^4)]X + [G_2 + G_3X +$$
$$X^8(G_{10} + G_{11}X + G_{14}X^4)]X^2 + [G_4 + G_5X + (G_6 + G_7X)X^2]X^4 +$$
$$G_{15}[(X^8X^4)(X^2X)].$$

Figure 1 gives the structure layout of the realization of the last expression for UDPLM for 4 inputs and 4 outputs.

Let us consider in more detail the possibilities of fast calculation of expressions (3). For the calculation of G_i the layout UDPLM itself may

be used. It is sufficiently to load into registres of preserving of G_i the values of the functions F_i for all i=1,...,2^m-1 and sequentially put into the inputs of layout values of $X=\alpha^{-k}$, where k=1,...,2^m-1. On the output we get the required values of G_i with the minimal time delay (2 ns for 1 G_k for one-cascade UDPLM). If for readjusing UDPLM from one function to another only a small number of values of outputs is changed, changes of the volume of the calculation for (3) will be drastically reduced. So by changing one value of the output at the point α_k from the old $F_{k_{old}}$ to the new $F_{k_{new}}$ expression (3) becomes as follows:

$$G_{i_{new}} = G_{i_{old}} + (F_{k_{new}} - F_{k_{old}})\alpha_k^{-i},$$

for all $i = 1, ..., 2^m - 1$.

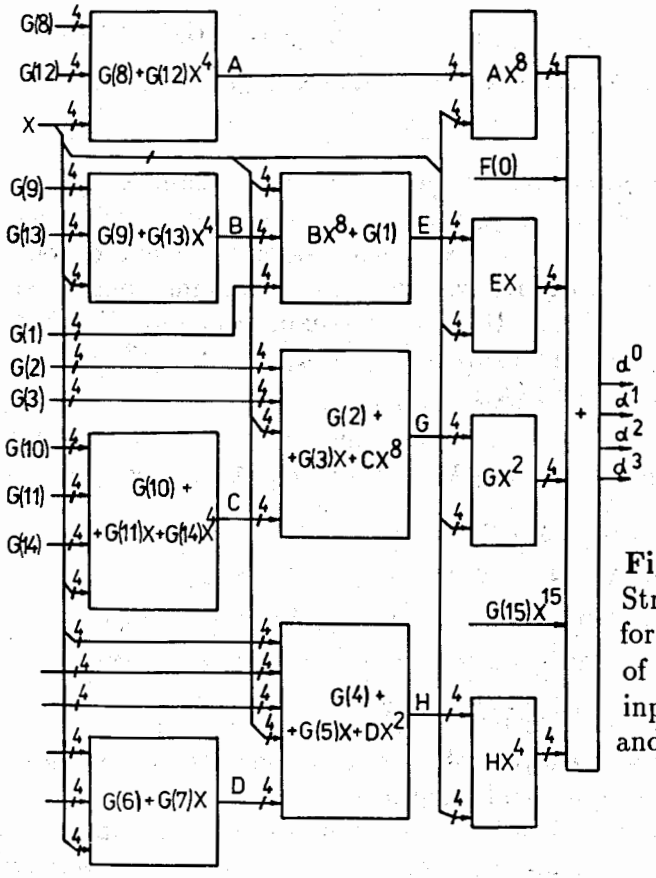

Figure 1. Structural scheme for realization of UDPLM for 4 inputs and 4 outputs

It is possible also to use systolic structures for the calculation of G_i. In this case the accumulation of G_i values is accomplished sequentially during 2^m-1 steps. The general structural scheme of operating one step taking 2^m-1 cycles of the systolic system is represented on figure 2. The calculation of expression (2) in more simple way than (3) is likely to be realized in the systolic system, for example, on the basis of the Gorner scheme. Though the speed of the calculations in the systolic systems is up to 5 billion operations per second as the process of calculations is iterative, for definite classes of the problems the delay time can be larger than is needed. In this case depending on the requirements on speed calculations may be parallel. For example, for m=6, $2^m-1=63$ it is possible to realize the calculation of F(X) as a sum F(X)=F1+F2+F3+F4. Here F1, F2, F3 and F4 are partial sums from (2) at i: from 0 to 15 for F1, from 16 to 31 for F2, from 32 to 47 for F3, and from 48 to 63 for F4. The calculation of F1, F2, F3 and F4 is performed parallelly, initial values of X^{16}, X^{32} and X^{48} for F2, F3 and F4 are calculated directly by means of the layouts similar to those given in appendix 1. Thus, for the calculation of F(X) instead of 63 cycles of the systolic system 16 cycles are required at 4-fold increasing of the hardware in the systolic system.

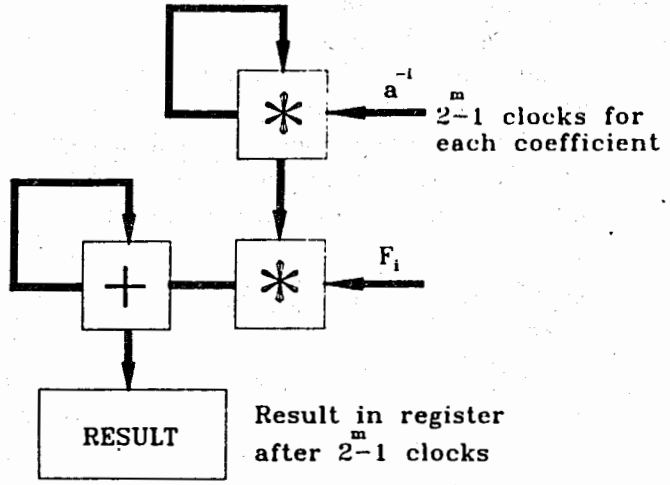

Figure 2. Structural scheme of getting values of coefficients G_i in systolic system. * is multiplication operation in Galois field, + is modulo 2 addition, F_i is the value of function at point α^i

4 Construction of m-input UDPLM by means of UDPLM of a lesser number of inputs

With increasing m the complexity of the expression for UDPLM increases, therefore it would be useful to get values of any switching function of m inputs by means of UDPLM of a lesser number of inputs/outputs, in other words, working in Galois fields of lesser order. Let us consider any function F with m inputs and m outputs. The domain of its definition X is a set of all binary of m length. Let Y be a domain of values of F (the length m binary number). We assume that the inputs (outputs) of the function are from left to right. Let us call for convenience m_1 left inputs (outputs) lower, the remaining $m_2 = m - m_1$ inputs (outputs) higher (we have $m > m_1 \geq m_2$). Partitioning X into $2^m \times 2$ classes K_i is as follows: element x from X belongs to K_i if its higher bits represent number i in binary form. Each class has $2^m \times 2$ elements. They differ only in lower bits. In each K_i let us define a pair of functions F_{i_1} and F_{i_2} as follows. Any x belonging to K_i may be represented in the form $(x_1, x_2, \ldots, x_{m1}, x_{m1+1}, \ldots, x_m)$, $(x_{m+1}, \ldots, x_m)$ being a constant for any x from K_i. If $y = F(x)$, then y can be also represented in the form $(y_1, y_2, \ldots, y_{m_1}, y_{m_1+1}, \ldots, y_m)$, and in accordance with the definition assume $F_{i_1}(x) = (y_1, \ldots, y_{m_1})$ and $F_{i_2} = (y_{m_1} + 1, \ldots, y_m)$. As for each K_i the high-order bits of inputs values are constant, we may take for any F_i as inputs only lower bits, and consequently to realize it in m_1 inputs/outputs of UDPLM. For each F_i let us calculate from expression (3) coefficients G_{ij}, where $i = \overline{1, 2^{m_2}}$, $j = \overline{1, 2^{m_1}}$. Then we may get the values of the function F (separately lower and higher bits) by formulae analogous to (2):

$$F = \sum_{k=1}^{2^{m_1}} G_k(X^k), \qquad G_k = \sum_{i=1}^{2^{m_2}} G_{k,i} p(i, x),$$

where $X = (x_1, \ldots, x_{m_1})$, $p(i, x)$ equal 1, when the higher bits of x coincide with i in the binary representation and equal zero otherwise. In other words, $p(i, x)$ is a term from all $(x_{m_1} + 1, \ldots, x_{m_2})$ and x_j is negated if in the j-th position of the number i in the binary representation is 0. To obtain all the bits of the outputs, 2 UDPLMs with m_1 inputs/outputs are necessary . The mode with UDPLM with m_1 inputs/outputs and UDPLM with m_2 inputs/outputs is possible.

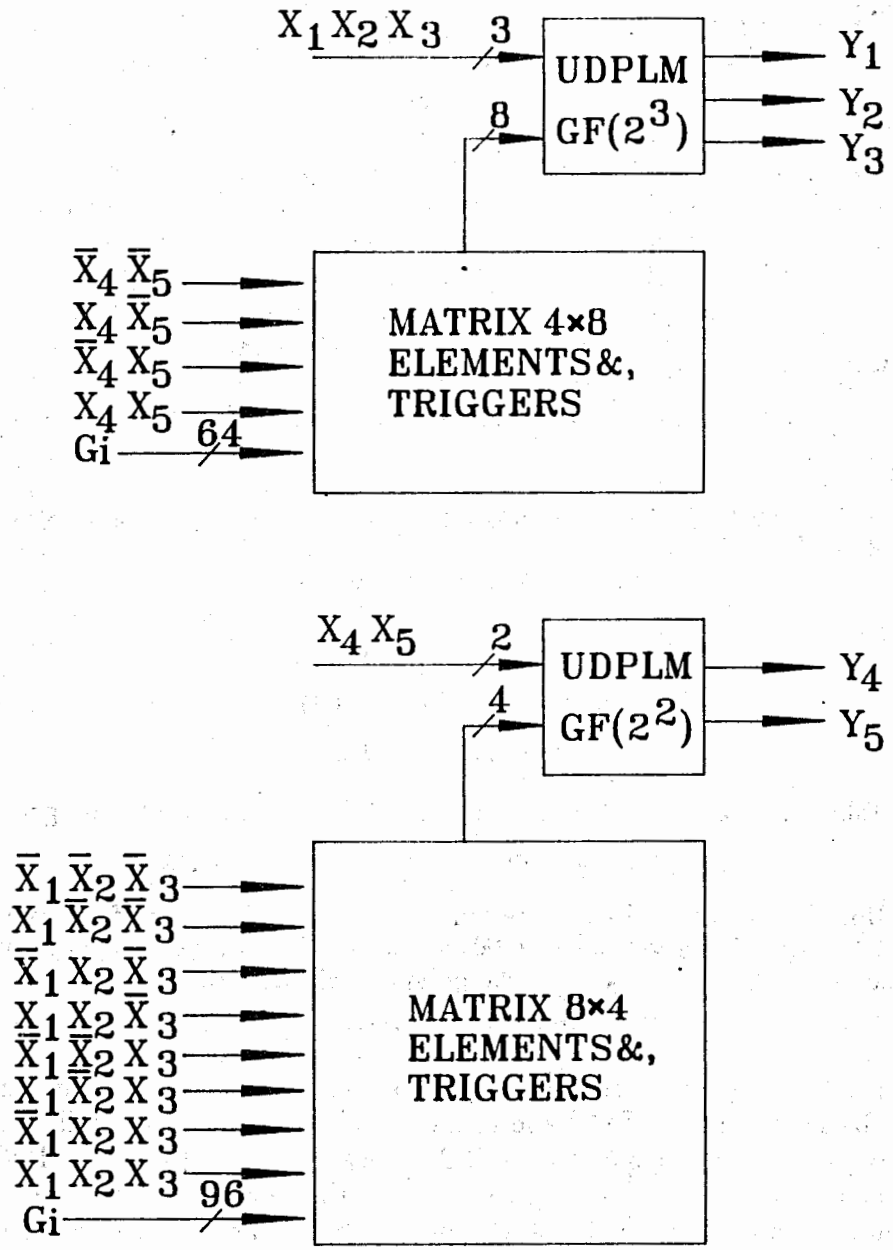

Figure 3. Structural scheme of UDPLM for $m = 5, m_1 = 3$ and $m_2 = 2$

Then G_{ij} from F_{i_1} is put into the first UDPLM, but G_{ij} from F_{i_2} is put into the second UDPLM (with m_2 inputs/outputs), besides for the second UDPLM the lower bits act as the higher ones and vice versa. In this case the memory capacity for storing the adjusting coefficients is equal to the storage for UDPLM for m inputs/outputs, but layouts for UDPLM themselves are simplier because of a lesser number of inputs/outputs. The global structural scheme for $m = 5, m_1 = 3$ and $m_2 = 2$ is represented in figure 3. It is obvious that the problem is simplified for a function with the number of outputs less than the number of inputs. In this case one can take m_1 equal a number of outputs, and UDPLM for obtaining F_{i_2} is not needed. So if the structural scheme in fig.3 for m=5 inputs had 3 outputs, 8×4 matrix for GF(2^2) would be absent.

EXAMPLE 2. It is necessary to construct a polynom for cortege of Boolean functions

$$f_0(X)f_1(X)f_2(X),$$

where

$$f_0(X) = (x_1 + 1)(x_3 + 1) + x_2,$$

$$f_1(X) = x_2 \bigvee x_1(x_3 + 1),$$

$$f_2(X) = x_1(x_2 \bigvee x_3).$$

Taking into account X^3=X+1 for the field GF(2^3) in (2),(3) we get

$$F(X) = \alpha X + \alpha^2 X^2 + \alpha X^3 + \alpha^4 X^4 + \alpha^4 X^5 + \alpha^2 X^6.$$

But if we partition by x_3, then in GF(2^2) for both values of x_3 we get a function f of 2 variables: $f(00)$=0, $f(01)$=0, $f(10)$=0, $f(11)$=1, and as in GF(2^2) $X^2 = X + 1$, the polynom will take the form:

$$F(X) = \alpha X + \alpha^2 X^2 + X^3.$$

It should be noted that $F(X) = a_0 a_1 \alpha^0$, where a_0, a_1 are Boolean variables both in the GF(2^3) and in the GF(2^2) cases when the realization of G_i is in the layout itself and $X = a_0\alpha^0 + a_1\alpha + a_2\alpha^2$ for GF(2^3) and $X = a_0\alpha^0 + a_1\alpha$ for GF(2^2). It is the case when we bring the layout to the level of the Zhegalkin polynom. This example was taken from example 1 in [20], where the global arithmetical polynom for this cortege of functions: $D(X) = 3x_1 + 3x_2 + x_1x_3$ has been obtained. In D(X) all operations are the operations of decimal arithmetics.

5 Construction of UDPLM for incompletely defined switching functions

In synthesis of switching functions we deal with a large important class of incompletely defined switching functions [21]. In works [22],[23] interesting results in the problem of constructing minimal polynomial forms for such functions have been obtained. However, in [22] the calculation of coefficients is connected with the solving of a system of linear equations. In [23] a class of polynomial forms is given with fast calculation of coefficients, but input limitations are imposed on the functions for which such forms can be obtained. Below it will be shown that for functions with a relatively large number of given values it is useful to use polynomial forms of Galois fields . There exist two directions. The first is to define the missing values of function as zeroes and to use the methods given in the above sections. This variant is not bad for a large number of problems. However, it is less preferable if a number of inputs is large (m=13 and larger), but a number of the defined values of function is much less than $2^m - 1$. Therefore, the second direction is to get an array of intermediate coefficients. It is more preferable if in UDPLM the class of inputs is fixed (and there are no limitations which inputs are fixed) but only the values of outputs of the switching function change. Let L values of function ($L \ll 2^m - 1$) be given. We will represent the function F(X) in the form:

$$F(X) = F(0) + \sum_{k=1}^{L} A_k X^k, \tag{4}$$

$$A_k = \sum_{j=1}^{L} K_{j_k}(F(j) + F(0)), \tag{5}$$

where the coefficients K_{j_k} depend only upon the array of inputs X. The matrix of the coefficients K_{j_k} is obtained by solving the matrix equation K×Y=E. All matrices have size L×L. E is a unitary matrix. K is a matrix of the required coefficients K_{j_k}, and Y is

$$Y = \begin{pmatrix} X_1 & X_2 & X_3 & \cdots & X_L \\ X_1^2 & X_2^2 & X_3^2 & \cdots & X_L^2 \\ X_1^3 & X_2^3 & X_3^3 & \cdots & X_L^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ X_1^L & X_2^L & X_3^L & \cdots & X_L^L \end{pmatrix},$$

where X_1,...,X_L are inputs of the switching function where it is defined. The equation is solved easily by standard algebraic methods, for

example, by the Gauss method, but with operations in the Galois field.

To prove the validity of (4)-(5) it is sufficient to substitute expression (5) for A_k into (4):

$$F(X) = F(0) + \sum_{k=1}^{2^m-1} X^k \Big(\sum_{j=1}^{2^m-1} K_{j_k}(F_{(j)} + F(0)) =$$

$$F(0) + \sum_{k=1}^{2^m-1} \sum_{j=1}^{2^m-1} (X^K (K_{j_k}(F_j + F(0))) =$$

$$F(0) + \sum_{j=1}^{2^m-1} (F_j + F(0)) \sum_{k=1}^{2^m-1} X^k K_{j_k}.$$

Finally, note that from K×Y=E it follows that the internal sum equals 0 for all $X \neq X_j$ and equals 1 for $X = X_j$. Here X_j is the input of the function on which its value equals F_j. It immediately follows that F(X)= F(0)+F_j+F(0)= F_j. So our proof is complete. Having the matrix K from (5) it is easy, and in systolic systems also, to get values of adjusting coefficients A_k used in (4). Polynom (4) itself is realized identically as calculations of polynom (2). The matrix K is made to save in cheaper external storage calculated and loaded, when necessary, adjusting coefficients A_k. In the given mode we made all the coefficients of the X degrees larger than L equal zero, therefore expressions for these degrees of X are not needed. As $L \ll 2^m - 1$, the obtained in such a mode layouts for this class of problems are more economic. Note that the equation A×Y=F should be solved directly (A is a vector of adjusting coefficients for (4)). However, in this case the obtaining of the adjusting coefficients, will require the solving of the system of linear equations for each new function. This leads to increasing delay time for obtaining the adjusting coefficients, which is not preferable for the class of problems under study.

6 Conclusion

As was shown, the representation of switching functions in the form of Galois field polynom is promising for the calculation and synthesis of UDPLM. As such modules are functionally complete, they can be implemented instead of EPROM. UDPLMs are purely combinatorial

ones and have no decodes, therefore, signal delays in them are minimal, though can be increased with increasing number of cascades. For obtaining maximal speed it is necessary to use one cascade and to realize it in the form of a chip. For loading the adjusting coefficients the programmable controlled interface should be installed additionally, using both systolic systems and UDPLM itself for increasing the speed of UDPLM readjusting.

APPENDIX 1.

Expressions of degrees of X in the basis of Galois field GF(2^4) with $X^4 = X + 1$

$$X = \alpha^0 \cdot 1 + \alpha^1 \cdot 2 + \alpha^2 \cdot 4 + \alpha^3 \cdot 8$$

$$X^2 = \alpha^0(1+4) + \alpha^1 \cdot 4 + \alpha^2(2+8) + \alpha^3 \cdot 8$$

$$X^3 = \alpha^0(1+5+6+a)+\alpha^1(3+5+8+c)+\alpha^2(3+4+5+6+9+a+c)+\alpha^3(2+4+8+a+c)$$

$$X^4 = \alpha^0(1+2+4+8) + \alpha^1(2+8) + \alpha^2(4+8) + \alpha^3 \cdot 8$$

$$X^5 = \alpha^0(1+3+4+5+8+9+c)+\alpha^1(2+4+6+9+a+c)+\alpha^2(2+4+6+9+a+c)$$

$$X^6 = \alpha^0(1+3+4+9+c)+\alpha^1(3+4+5+6+9+a+c)+\alpha^2(2+3+4+5+a)+$$
$$\alpha^3(2+4+8+a+c)$$

$$X^7 = \alpha^0(1+2+3+4+7+a+b+e)+\alpha^1(2+3+5+a+c+d+e)+$$
$$\alpha^2(3+5+6+8+a+b)+\alpha^3(2+4+8+9+a+c+e)$$

$$X^8 = \alpha^0(1+2)+\alpha^1(4+8)+\alpha^2 \cdot 2+\alpha^3 \cdot 8$$

$$X^9 = \alpha^0(1+3+6+8+a+c)+\alpha^1(2+3+5+6+8+9)+\alpha^2(3+5+8+c)+\alpha^3(2+4+8+a+c)$$

$$X^{10} = \alpha^0(1+2+3+5+6+8+a)+\alpha^1(2+4+6+9+a+c)+\alpha^2(2+4+6+9+a+c)$$

$$X^{11} = \alpha^0(1+4+5+7+b+c+d)+\alpha^1(2+3+4+5+6+9+b+c+e)+$$
$$\alpha^2(2+3+5+a+c+d+e)+\alpha^3(2+4+8+9+a+c+e)$$

$$X^{12} = \alpha^0(1+2+5+9+a+c)+\alpha^1(2+3+4+5+a)+\alpha^2(2+3+5+6+8+9)+$$
$$\alpha^3(2+4+8+a+c)$$

$$X^{13} = \alpha^0(1+2+3+6+7+9+d+e)+\alpha^1(3+4+5+8+9+d)+$$
$$\alpha^2(2+3+4+5+6+9+b+c+e)+\alpha^3(2+4+8+9+a+c+e)$$

$$X^{14} = \alpha^0(1+2+4+5+6+7+8+e)+\alpha^1(3+5+6+8+a+b)+$$
$$\alpha^2(3+4+5+8+9+d)+\alpha^3(2+4+8+9+a+c+e)$$

$$X^{15} = \alpha^0(1+2+3+4+5+6+7+8+9+a+b+c+d+e+f)$$

Compression form of recording the polinom's terms is used. The coefficients of α^i are represented in binary-position form: in hex numbers positions they show which a_i is in the term, for example, number $5 = 0101_{16} = a_0a_2$, and $d = 1101_{16} = a_0a_2a_3$.

APPENDIX 2.

Expressions of multiplication of coefficients G by degrees of X in the Galois field GF(2^4) with $X^4 = X + 1$

$$GX = \alpha^0(1\cdot 1 + 2\cdot 8 + 4\cdot 4 + 8\cdot 2) + \alpha^1(1\cdot 2 + 2(1+8) + 4(4+8) + 8(2+4)) +$$
$$\alpha^2(1\cdot 4 + 2\cdot 2 + 4(1+8) + 8(4+8)) + \alpha^3(1\cdot 8 + 2\cdot 4 + 4\cdot 2 + 8(1+8))$$

$$GX^2 = \alpha^0(1(1+4) + 2\cdot 8 + 4(2+8) + 8\cdot 4) + \alpha^1(1\cdot 4 + 2(1+4+8) + 4\cdot 2 +$$
$$8(4+2+8)) + \alpha^2(1(2+8) + 2\cdot 4 + 4(1+4+8) + 8\cdot 2) + \alpha^3(1\cdot 8 + 2(2+8) + 4\cdot 4 + 8(1+4+8))$$

$$GX^3 = \alpha^0(1(1+5+6+a) + 2(2+4+8+a+c) + 4(3+4+5+6+9+a+c) +$$
$$8(3+5+8+c)) + \alpha^1(1(3+5+8+c) + 2(1+5+6+2+4+8+c) +$$
$$4(3+5+6+9+2+8) + 8(8+4+6+9+a) + \alpha^2(1(3+4+5+6+9+a+c) +$$
$$2(3+5+8+c) + 4(1+5+6+2+4+8+c) + 8(3+5+6+9+2+8)) +$$
$$\alpha^3(1(2+4+8+a+c) + 2(3+4+5+6+9+a+c) + 4(3+5+8+c) + 8(1+5+6+2+4+8+c))$$

$$GX^4 = \alpha^0(1(1+2+4+8) + 2\cdot 8 + 4(4+8) + 8(2+8)) + \alpha^1(1(2+8) + 2(1+2+4) +$$
$$4\cdot 4 + 8(2+4) + \alpha^2(1(4+8) + 2(2+8) + 4(1+2+4) + 8\cdot 4) +$$
$$\alpha^3(1\cdot 8 + 2(4+8) + 4(2+8) + 8(1+2+4))$$

$$GX^5 = \alpha^0(1(1+3+4+5+8+9+c) + 4(2+4+6+9+a+c) + 8(2+4+6+9+a+c)) +$$
$$\alpha^1(1(2+4+6+9+a+c) + 2(1+3+4+5+8+9+c) + 4(2+4+6+9+a+c)) +$$
$$\alpha^2(1(2+4+6+9+a+c) + (2(2+4+6+9+a+c) + 4(1+3+4+5+8+9+c) +$$
$$8(2+4+6+9+a+c)) +$$
$$\alpha^3(2(2+4+6+9+a+c) + 4(2+4+6+9+a+c) + 8(1+3+4+5+8+9+c))$$

$$GX^6 = \alpha^0(1(1+3+4+9+c) + 2(2+4+8+a+c) + 4(2+3+4+5+a) + 8(3+4+5+6+9+a+$$
$$c)) + \alpha^1(1(3+4+5+6+9+a+c) + 2(1+3+9+2+8+a) + 4(3+5+8+c) +$$
$$8(6+9+c+2)) +$$
$$\alpha^2(1(2+3+4+5+a) + 2(3+4+5+6+9+a+c) + 4(1+3+9+2+8+a) +$$
$$8(3+5+8+c)) +$$
$$\alpha^3(1(2+4+8+a+c) + 2(2+3+4+5+a) + 4(3+4+5+6+9+a+c) +$$

$$8(1+3+9+2+8+a))$$

$$GX^7 = \alpha^0(1(1+2+3+4+7+a+b+e) + 2(2+4+8+9+a+c+e) +$$
$$4(3+5+6+8+a+b) + 8(2+3+5+a+c+d+e)) +$$
$$\alpha^1(1(2+3+5+a+c+d+e) +$$
$$2(1+3+7+b+8+9+c) + 4(3+5+6+b+2+4+9+c+e) + 8(2+c+d+e+6+8+b)) +$$
$$\alpha^2(1(3+5+6+8+a+b) + 2(2+3+5+a+c+d+e) + 4(1+3+7+b+8+9+c) +$$
$$8(3+5+6+b+2+4+9+c+e)) + \alpha^3(1(2+4+8+9+a+c+e) +$$
$$2(3+5+6+8+a+b) + 4(2+3+5+a+c+d+e) + 8(1+3+7+b+8+9+c))$$

$$GX^8 = \alpha^0(1(1+2) + 2\cdot 8 + 4\cdot 2 + 8(4+8) + \alpha^1(1(4+8) + 2(1+2+8) +$$
$$4(2+8) + 8(4+8+2) + \alpha^2(1\cdot 2 + 2(4+8) + 4(1+2+8) + 8(2+8) +$$
$$\alpha^3(1\cdot 8 + 2\cdot 2 + 4(4+8) + 8(1+2+8))$$

$$GX^9 = \alpha^0(1(1+3+6+8+a+c) + 2(2+4+8+a+c) + 4(3+5+8+c) + 8(2+3+5+6+8+9))$$
$$+\alpha^1(1(2+3+5+6+8+9) + 2(1+3+6+2+4) + 4(3+5+2+4+a) + 8(2+6+9+c))$$
$$+\alpha^2(1(3+5+8+c) + 2(2+3+5+6+8+9) + 4(1+3+6+2+4) + 8(3+5+2+4+a)) +$$
$$\alpha^3(1(2+4+8+a+c) + 2(3+5+8+c) + 4(2+3+5+6+8+9) + 8(1+3+6+2+4))$$

$$GX^{10} = \alpha^0(1(1+2+3+5+6+8+a) + 4(2+4+6+9+a+c) + 8(2+4+6+9+a+c)) +$$
$$\alpha^1(1(2+4+6+9+a+c) + 2(1+2+3+5+6+8+a) + 4(2+4+6+9+a+c) +$$
$$\alpha^2(1(2+4+6+9+a+c) + 2(2+4+6+9+a+c) + 4(1+2+3+5+6+8+a) +$$
$$8(2+4+6+9+a+c)) +$$
$$\alpha^3(2(2+4+6+9+a+c) + 4(2+4+6+9+a+c) + 8(1+2+3+5+6+8+a))$$

$$GX^{11} = \alpha^0(1(1+4+5+7+b+c+d) + 2(2+4+8+9+a+c+e) + 4(2+3+5+a+c+d+e) +$$
$$8(2+3+4+5+6+9+b+c+e)) + \alpha^1(1(2+3+4+5+6+9+b+c+e) +$$
$$2(1+5+7+b+d+2+8+9+a+e) + 4(3+5+d+4+8+9) + 8(4+6+9+b+a+d)) +$$
$$\alpha^2(1(2+3+5+a+c+d+e) + 2(2+3+4+5+6+9+b+c+e) +$$
$$4(1+4+5+7+b+c+d+2+4+8+9+a+c+e) + 8(3+5+d+4+8+9)) +$$
$$\alpha^3(1(2+4+8+9+a+c+e) + 2(2+3+5+a+c+d+e) +$$
$$4(2+3+4+5+6+9+b+c+e) + 8(1+5+7+b+d+2+8+9+a+e))$$

$$GX^{12} = \alpha^0(1(1+2+5+9+a+c) + 2(2+4+8+a+c) + 4(2+3+5+6+8+9) + 8(2+3+4+5$$
$$+a)) + \alpha^1(1(2+3+4+5+a) + 2(1+5+9+4+8) + 4(2+3+5+6+8+9+$$
$$4+a+c) + 8(4+a+6+8+9)) +$$

$$\alpha^2(1(2+3+5+6+8+9)+2(2+3+4+5)+4(1+5+9+4+8)+$$
$$8\cdot(3+5+6+9+4+a+c))+$$
$$\alpha^3(1(2+4+8+a+c)+2(2+3+5+6+8+9)+4(2+3+4+5+a)+$$
$$8(1+5+9+4+8))$$

$$GX^{13}=\alpha^0(1(1+2+3+6+7+9+d+e)+2(2+4+8+9+a+c+e)+$$
$$4(2+3+4+5+6+9+b+c+e)+$$
$$8(3+4+5+8+9+d))+\alpha^1(1(3+4+5+8+9+d)+$$
$$2(1+3+6+7+d+4+8+a+c)+$$
$$4(2+3+4+5+6+9+b+c+e+2+4+8+9+a+c+e)+8(8+d+2+6+b+c+e))+$$
$$\alpha^2(1(2+3+4+5+6+9+b+c+e)+2(3+4+5+8+9+d)+$$
$$4(1+3+6+7+d+4+8+a+c)+$$
$$8(3+5+6+b+8+a))+\alpha^3(1(2+4+8+9+a+c+e)+$$
$$2(2+3+4+5+6+9+b+c+e)+$$
$$4(3+4+5+8+9+d)+8(1+3+6+7+d+4+8+a+c))$$

$$GX^{14}=\alpha^0(1(1+2+4+5+6+7+8+e)+2(2+4+8+9+a+c+e)+$$
$$4(3+4+5+8+9+d)+$$
$$8(3+5+6+8+a+b)+\alpha^1(1(3+5+6+8+a+b)+2(1+5+6+7+9+a+c)+$$
$$4(3+5+d+2+a+c+e)+8(6+a+b+4+9+d)+\alpha^2(1(3+4+5+8+9+d)+$$
$$2(3+5+6+8+a+b)+4(1+5+6+7+9+a+c)+8(3+5+d+2+a+c+e)+$$
$$\alpha^3(1(2+4+8+9+a+c+e)+2(3+4+5+8+9+d)+4(3+5+6+8+a+b)+$$
$$8(1+5+6+7+9+a+c))$$

References

1. D.A.Pospelov. Logicheskie metody analiza i sinteza skhem. M.:Energiya, 1974.

2. G.S.Avsarkisyan, G.S.Braylovskiy. Predstavlenie logicheskikh funktsiy v vide polinomov Zhegalkina.- Avtomatika i vychislitelnaya tekhnika, 1975, No.6, pp.6-8.

3. V.D.Malyugin. Realizatsiya bulevyh funktsiy arifmeticheskimi polinomami.- Avtomatika i telemekhanika, 1982, No.4, pp.84-93.

4. V.D.Malyugin. Realizatsiya kortezhey bulevykh funktsiy posredstvom lineynykh arifmeticheskikh polinomov. - Avtomatika i telemekhanika, 1984, No.2, pp.114-122.

5. K.S.Menger. A Transform for Logic Networks. - IEEE Transactions on COMPUTERS,1969, vol.C-18, No.3, pp.241-250.

6. B.Benjauthrit, S.Reed. Galois Switching Functions and their Applications. - IEEE Transactions on COMPUTERS, 1976, vol.C-25, No.1, pp.78-86.

7. W.R.English. Synthesis of Finite State Algorithms in a Galois GF(p^n). - IEEE Transactions on COMPUTERS, 1981, vol.C-30, No.3, pp.225-229.

8. I.N.Aleksandrov, R.I.Gaydamaka, N.M.Nikityuk, V.P.Shirikov. Raschet perekluchatelnykh funktsiy, predstavlennykh elementami polya Galua GF(2^m). - Preprint JINR P10-84-865, Dubna, 1984.

9. U.Piterson, A.Waldon. Kody, ispravlyayushie oshibki. M.:Mir, 1976.

10. N.M.Nikityuk. Metod sindromnogo kodirovaniya i ego primenenie dlya bystrogo apparatnogo otbora sobytiy na osnove protsessorov, operiruyushikh v pole Galua GF(2^m). - Preprint JINR P11-80-484, Dubna, 1980.

11. N.M.Nikityuk. Voprosy optimalnogo kodirovaniya v godoskopicheskikh sistemakh. - Pribory i tekhnika eksperimenta, 1983, No.3, pp.74-81.

12. I.N.Aleksandrov, R.I.Gaydamaka, N.M.Nikityuk. Primenenie analiticheskikh vychisleniy dlya rascheta logicheskikh skhem i spetsprocessorov. - V knige "Analiticheskie vychisleniya na EVM i ikh primenenie v teoreticheskoy fizike", Trudy Mezhdunarodnogo soveshaniya, Dubna, JINR, 1985, pp.295-300.

13. C.S.Yeh, I.S.Reed, T.K.Truong. Systolic Multipliers for Finite Fields GF(2^m). - IEEE Transactions on COMPUTERS, 1984, vol.C-33, No.4, pp.357-360.

14. C.C.Wang, T.K.Truong, H.M.Shao, L.J.Deutsch, J.K.Omura, I.S.Reed. VLSI Architectures for Computing Multiplications and Inverses in GF(2^m).- IEEE Transactions on COMPUTERS, 1985, vol.C-34, No.8, pp.709-717.

15. N.M.Nikityuk. Sovmeshennye operatsii v pole Galua GF(2^m) i ikh primenenie. - Preprint JINR P11-87-54, Dubna, 1987. .

16. I.S. Hsu, T.K.Truong, L.J.Deutsch, I.S.Reed. A Comparison of VLSI Architecture of Finite Field Multipliers Using a Dual, Normal, or Standard Bases. - IEEE Transactions on COMPUTERS, 1988, vol.37, No.6, pp.735-739.

17. C.C.Wang. An Algorithm to Design Finite Field Multipliers Using a Self-Dual Normal Basis. - IEEE Transactions on COMPUTERS, 1989, vol.38, No.10, pp.1457-1460.

18. A.Pincin. A New Algorithm for Multiplication on Finite Fields. - IEEE Transactions on COMPUTERS, 1989, vol.38, No.7, pp.1045-1049.

19. N.M.Nikityuk. Bystriy algoritm dlya vypolneniya operatsiy umnozheniya v pole Galua GF(2^m). - Upravlyayushie sistemy i mashiny, 1990, No.6, pp.21-28.

20. V.D.Malyugin, G.A.Kukharev, V.P.Shmerko. Preobrazovaniya polinomialnykh form bulevykh funktsiy. - Preprint Instituta problem upravleniya, Moskva, 1986.

21. R.K.Breyton, G.D.Hatchel, A.L.Sandjovani-Vinchitelli. Sintez mnogourovnevyh kombinatsionnykh logicheskikh skhem. - Trudy Instituta inzhenerov po elektronike i radioelektronike, 1990, tom 78, No.2, pp.38-83.

22. G.S.Avsarkisyan. Polinomialnye formy chastichnykh bulevykh funktsiy i nekotorye ikh prilozheniya. - Izvestiya AN SSSR. Tekhnicheskaya kibernetika, 1983, No.5, pp.50-58.

23. G.S.Avsarkisyan. Rekurentnye polinomialnye formy chastichnykh bulevyh funktsiy. - Izvestiya AN SSSR. Tekhnicheskaya kibernetika, 1987, No.4, pp.131-135.

Александров И.Н., Котов В.М., Никитюк Н.М. E10-93-412

Некоторые вопросы применения
переключательных функций в полях Галуа GF(2^m)

Рассмотрена возможность применения полиномиальных форм построения переключательных функций в полях Галуа GF(2^m), показана перспективность их использования при синтезе универсальных динамически программируемых логических модулей. Рассмотрены варианты синтеза схем как для полностью, так и не полностью определенных функций. Приведен пример универсального динамически программируемого модуля 4-х переменных.

Работа выполнена в Лаборатории высоких энергий ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна, 1993

Aleksandrov I.N., Kotov V.M., Nikityuk N.M. E10-93-412

Some Questions of an Application
of Galois Fields GF(2^m) Switching Functions

Possibility of an application of polynomial forms of constructing Galois fields GF(2^m) switching functions is considered, the perspectivity of their using for synthesis of universal dynamically programmable logic modules (UDPLM) is shown. Modes of layouts synthesis both for completely and incompletely defined functions are presented. An example of the universal dynamically programmable logic module of 4 variables is given.

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

Preprint of the Joint Institute for Nuclear Research. Dubna, 1993