ISSN 0146-4116, Automatic Control and Computer Sciences, 2010, Vol. 44, No. 3, pp. 171–177. © Allerton Press, Inc., 2010.
Original Russian Text © V.P. Suprun, D.A. Gorodetskii, 2010, published in Avtomatika i Vychislitel'naya Tekhnika, 2010, No. 3, pp. 72–@@.

Synthesis of *n*-operand Modulo-Three Adders

V. P. Suprun and D. A. Gorodetskii

Belarusian State University, pr. Nezavisimosti 4, Minsk, 220030 Belarus
e-mail: suprun@bsu.by, danila.gorodecky@gmail.com

Received February 11, 2010

Abstract—A method of synthesis of modulo-three adders for the case of data representation in positional codes is presented. The method is oriented to the construction of two-level circuits consisting of OR elements and exclusive OR elements with given thresholds. The method is generalized to the realization of the operation $\pm X_1 \pm X_2 \ldots \pm X_n = S \pmod 3$. A logical circuit of an adder with a controlled input is suggested such that the value of one of the digits of the result of the summation is realized on its single output.

Key words: modular arithmetic, symmetric Boolean functions, logical circuit, adder, adder with a controlled input

DOI: 10.3103/S0146411610030089

1. INTRODUCTION

It is known [1, 2] that the use of modular arithmetic allows us to significantly raise the velocity of discrete information processing owing to the parallelization of the processing procedure. This parallelization frequently depends on the depth of the logical circuit. The apparatus of modular arithmetic is widely used, in particular, when designing systems based on crystals. Hence, the increased effectiveness of such applications centers around the development of libraries for computing devices that contain modular arithmetic equipment (designed for the realization of basic arithmetical operations, including summation over the given module) [2, 3].

By now, a number of methods for the synthesis of modular arithmetic equipment are known. Among them we note the following: in [4], a method for the synthesis of adders based on the application of the principle of local coding for symmetric Boolean functions is presented taking into account the commutativity of the summation operation; the method of block-structural synthesis [5] is oriented to the synthesis of devices meant for the calculation of arithmetical operations that are superpositions of the summation and multiplication operations over the given module. Patents for invention of the Belorussia Republic [6–9] have been obtained for some logical circuits of modulo-three adders synthesized using the methods from [4, 5].

In the present paper, a problem of synthesis of *n*-operand module-three adders is considered provided that the data are presented in a positional code. The development of the analytical formulas for the assignment of the function for the adder's output is based on the properties of the application of symmetric Boolean functions. The schematic realization of the adder expects the use of the logical elements OR and exclusive OR with various thresholds.

2. MAIN NOTIONS AND DEFINITION

Let $C(n, 3)$ be an adder designed for the calculation (realization) of the arithmetic operation $X_1 + X_2 + \ldots + X_n = S \pmod 3$ with the condition that the input operands $X_1, X_2, \ldots, X_n$ and the output operand S are presented in positional codes, i.e.,

$$X_1 = 2x_2^1 + x_1^1, \quad X_2 = 2x_2^2 + x_1^{2(\,)}, \ldots, X_n = 2x_2^n + x_1^n, \quad \text{and} \quad S = 2s_2 + s_1,$$

where $X_1, X_2, \ldots, X_n, S \in \{0, 1, 2\}$ and $x_1^1,\ x_2^1,\ x_1^2,\ x_2^2,\ \ldots,\ x_1^n,\ x_2^n,\ s_1, s_2 \in \{0, 1\}$.

The adder $C(n, 3)$ has $2n$ inputs, such that the values of the input variables $x_1^1,\ x_2^1,\ x_1^2,\ x_2^2,\ ...,\ x_1^n,\ x_2^n$ are fed to it, and two outputs, such that the values of the low-order digit s_1 and the high-order digit s_2 of the result of the implementation of the adding operation

$$X_1 + X_2 + ... + X_n = S \pmod 3$$

are realized on them.

As usual, a symmetric Boolean function $F = F(x_1, x_2, \ldots, x_n)$ is considered to mean a function that does not change its value with any permutation of the variables. The main property of the function F is as follows: if the function F takes the unit value on any tuple of values of n variables that contains $a (0 \le a \le n)$ units, then the function F also takes the value equal to 1 on any other tuple with the same number a of units. Such a value a is said to be an *effective number* of the function F. In the general case, a symmetric Boolean function $F = F(x_1, x_2, \ldots, x_n)$ is liable to have r effective numbers, where $0 \le r \le n + 1$. If $r = 1$, then a symmetric Boolean function is said to be *elementary* (or *basic*) and is denoted by $F = F_n^a\ (x_1, x_2, \ldots, x_n)$.

The element EXCLUSIVE OR with a threshold a such that the values of the variables $x_1, x_2, ..., x_n$ are fed to its n inputs realizes the function $F = F_n^a(x_1, x_2, \ldots, x_n)$.

3. ANALYTIC REPRESENTATION OF THE LOGICAL FUNCTIONS S_1 AND S_2

Analytic representations in the form of a disjunction of elementary symmetric Boolean functions depending on $3n$ variables are valid for the logical functions S_1 and S_2.

Theorem 1. If $n \ge 2$, $X^1 = \{x_1^1,\ x_1^2,\ ...,\ x_1^n\}$, $X^2 = \{x_2^1,\ x_2^2,\ ...,\ x_2^n\}$, then

$$S_1(X^1, X^2) = \bigvee_{i=1}^{k_1} F_{3n}^{3i-2}(X^1, X^2, X^2), \tag{1}$$

$$S_2(X^1, X^2) = \bigvee_{j=1}^{k_2} F_{3n}^{3j-1}(X^1, X^2, X^2), \tag{2}$$

where $F = F_{3n}^{3i-2}(X^1, X^2, X^2)$ and $F = F_{3n}^{3j-1}(X^1, X^2, X^2)$ are elementary symmetric Boolean functions depending on the $3n$ variables $x_1^1,\ x_2^1,\ x_1^1,\ x_1^2,\ x_2^2,\ x_2^2, ..., x_1^n,\ x_2^n,\ x_2^n$, whose effective numbers are equal to $3i - 2$ and $3j - 1$, respectively, $k_1 = \left[\frac{2n+2}{3}\right]$, $k_2 = \left[\frac{2n+1}{3}\right]$, $i = 1, 2, ..., k_1$ and $j = 1, 2, ..., k_2$.

Proof. Let us denote $S^* = X_1 + X_2 + ... + X_n$, ($S^*$ is the arithmetic sum of n numbers). As $X_1, X_2, ..., X_n$, $S \in \{0, 1, 2\}$, then we have $S^* \in \{0, 1, 2, ..., 2n\}$.

Since $S^* = S \pmod 3$ and $S = 2s_2 + s_1$, the functions S_1 and S_2 take the unit values if and only if $S^* \in \{1, 4, ..., 3k_1 - 2\}$ and $S^* \in \{2, 5, ..., 3k_2 - 1\}$, respectively, where $k_1 = \left[\frac{2n+2}{3}\right]$ and $k_2 = \left[\frac{2n+1}{3}\right]$.

It follows from here that

$$S_1 = \begin{cases} 1, & \text{if} \quad X_1 + X_2 + \ldots + X_n = 3i - 2; \\ 0, & \text{otherwise}, \end{cases}$$

$$S_2 = \begin{cases} 1, & \text{if} \quad X_1 + X_2 + \ldots + X_n = 3j - 1; \\ 0, & \text{otherwise} \end{cases}$$

or

$$S_1 = \begin{cases} 1, & \text{if} \quad 2x_2^1 + x_1^1 + 2x_2^2 + x_1^2 + \ldots + 2x_2^n + x_1^n = 3i - 2; \\ 0, & \text{otherwise,} \end{cases}$$

$$S_2 = \begin{cases} 1, & \text{if} \quad 2x_2^1 + x_1^1 + 2x_2^2 + x_1^2 + \ldots + 2x_2^n + x_1^n = 3j - 1; \\ 0, & \text{otherwise,} \end{cases}$$

where $i = 1, 2, \ldots, k_1$ and $j = 1, 2, \ldots, k_2$.

The above representations of the logical functions S_1 and S_2 are equivalent to corresponding formulas (1) and (2).

Let us consider some special cases of the use of Theorem 1.

1) If $n = 2$, then formulas (1) and (2) are in the form

$$S_1(x_1^1, x_1^2, x_2^1, x_2^2) = F_6^1(x_1^1, x_1^2, x_2^1, x_2^1, x_2^2, x_2^2) \vee F_6^4(x_1^1, x_1^2, x_2^1, x_2^1, x_2^2, x_2^2),$$

$$S_2(x_1^1, x_1^2, x_2^1, x_2^2) = F_6^2(x_1^1, x_1^2, x_2^1, x_2^1, x_2^2, x_2^2).$$

2) If $n = 3$, then formulas (1) and (2) take the form

$$S_1(x_1^1, x_1^2, x_1^3, x_2^1, x_2^2, x_2^3) = F_9^1(x_1^1, x_1^2, x_1^3, x_2^1, x_2^1, x_2^2, x_2^2, x_2^3, x_2^3) \vee F_9^4(x_1^1, x_1^2, x_1^3, x_2^1, x_2^1, x_2^2, x_2^2, x_2^3, x_2^3),$$

$$S_2(x_1^1, x_1^2, x_1^3, x_2^1, x_2^2, x_2^3) = F_9^2(x_1^1, x_1^2, x_1^3, x_2^1, x_2^1, x_2^2, x_2^2, x_2^3, x_2^3) \vee F_9^5(x_1^1, x_1^2, x_1^3, x_2^1, x_2^1, x_2^2, x_2^2, x_2^3, x_2^3).$$

3) If $n = 4$, then formulas (1) and (2) take the form

$$\begin{aligned} S_1(x_1^1, x_1^2, x_1^3, x_1^4, x_2^1, x_2^2, x_2^3, x_2^4) = {} & F_{12}^1(x_1^1, x_1^2, x_1^3, x_1^4, x_2^1, x_2^2, x_2^2, x_2^3, x_2^3, x_2^4, x_2^4) \\ & \vee F_{12}^4(x_1^1, x_1^2, x_1^3, x_1^4, x_2^1, x_2^1, x_2^2, x_2^2, x_2^3, x_2^3, x_2^4, x_2^4) \\ & \vee F_{12}^7(x_1^1, x_1^2, x_1^3, x_1^4, x_2^1, x_2^1, x_2^2, x_2^2, x_2^3, x_2^3, x_2^4, x_2^4), \end{aligned}$$

$$\begin{aligned} S_2(x_1^1, x_1^2, x_1^3, x_1^4, x_2^1, x_2^2, x_2^3, x_2^4) = {} & F_{12}^2(x_1^1, x_1^2, x_1^3, x_1^4, x_2^1, x_2^1, x_2^2, x_2^2, x_2^3, x_2^3, x_2^4, x_2^4) \\ & \vee F_{12}^5(x_1^1, x_1^2, x_1^3, x_1^4, x_2^1, x_2^1, x_2^2, x_2^2, x_2^3, x_2^3, x_2^4, x_2^4) \\ & \vee F_{12}^8(x_1^1, x_1^2, x_1^3, x_1^4, x_2^1, x_2^1, x_2^2, x_2^2, x_2^3, x_2^3, x_2^4, x_2^4). \end{aligned}$$

Theorem 1 cited above is used when describing the method of adders $C(n, 3)$ synthesis on the condition that the input and the output operands are presented in positional codes.

4. METHOD FOR THE SYNTHESIS OF $C(n, 3)$ ADDERS

The method for the synthesis of $C(n, 3)$ adders is oriented to the application of the logical elements OR and exclusive OR with a given threshold. Logical circuites of $C(n, 3)$ adders synthesized on the basis of the use of formulas (1) and (2) will consist of two parts. These parts are the logical subschemes $C_1(n, 3)$ and $C_2(n, 3)$ on the unit output of which the logical functions S_1 and S_2, respectively, are realized.

The logical subscheme $C_1(n, 3)$ is synthesized by formula (1) immediately and logical subscheme $C_2(n, 3)$ synthesized by formula (2). Both subschemes have two levels. The EXCLUSIVE OR elements with a given threshold form the first levels of the logical circuites $C_1(n, 3)$ and $C_2(n, 3)$. The second levels are formed by the logical elements OR.

It should be noted as an exclusion that, if $n = 2$, then the first level of the subschema $C_2(2, 3)$ contains only one EXCLUSIVE OR element with the threshold 2, whereas the element OR is lacking.

The first level of the logical subschema $C_1(n, 3)$ contains k_1 EXCLUSIVE OR elements $A_1, A_2, \ldots, A_{k_1}$, where $k_1 = \left[\frac{2n+2}{3}\right]$. At that, the threshold p of the logical element A_i is calculated by the formula $p(A_i) = 3i - 2$, where $i = 1, 2, \ldots, k_1$.

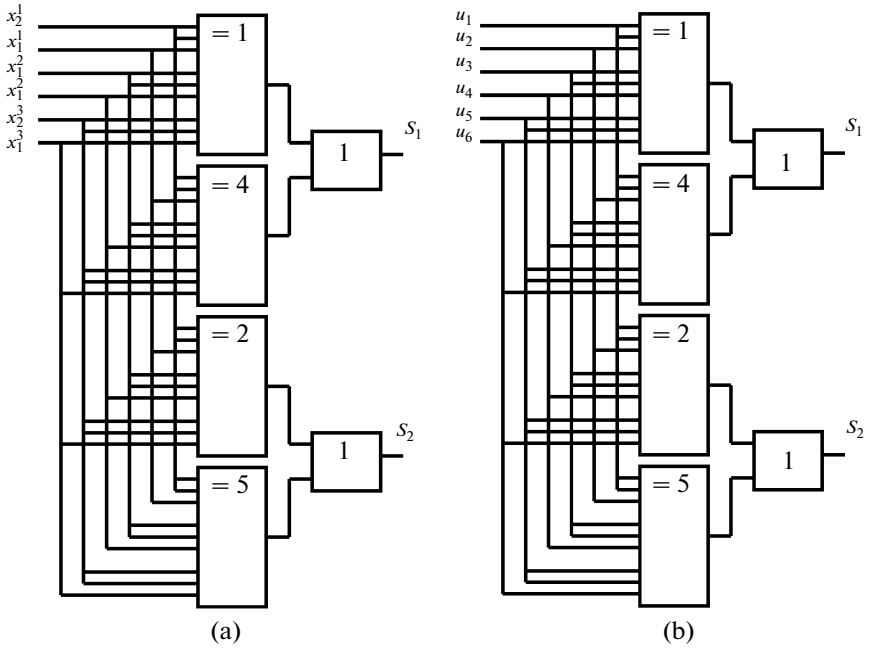

Fig. 1. Logical circuits of the adders: (a) $C(3, 3)$; (b) $C^*(3, 3)$.

The first level of the logical subschema $C_2(n, 3)$ contains k_2 EXCLUSIVE OR elements B_1, B_2, ..., B_{k_2}, where $k_2 = \left[\frac{2n+1}{3}\right]$. At that, the threshold p of the logical element B_j is calculated by the formula $p(B_j) = 3j - 1$, where $j = 1, 2, ..., k_2$.

Each of the logical EXCLUSIVE OR elements A_1, A_2, ..., A_{k_1}, and B_1, B_2, ..., B_{k_2} has $3n$ digits on which the values of the low-order digits x_1^1, x_1^2, ..., x_1^n and twice the values of the high-order digits x_2^1, x_2^2, ..., x_2^n of the operands X_1, X_2, ..., X_n come.

It is appropriate to mark the fact that the schemes $C_1(n, 3)$ and $C_2(n, 3)$ contain the same number of logical elements equal to $k_1 = k_2$ on the condition that $n = 3m$ or $n = 3m + 1$, where m is a natural number. If $n = 3m - 1$, then $k_1 = k_2 + 1$; i.e., the scheme $C_1(n, 3)$ has one more logical element as compared with the scheme $C_2(n, 3)$.

It follows from the method described that the logical circuit of the $C(n, 3)$ adder contains $\left[\frac{2n+2}{3}\right] + \left[\frac{2n+1}{3}\right]$ logical EXCLUSIVE OR elements with different thresholds and two OR elements. At that, the complicity L (by the number of inputs of logical elements) of the logical circuit of the $C(n, 3)$ adder is calculated by the formula

$$L(C(n, 3)) = k_1(3n + 1) + k_2(3n + 1) = (3n + 1)\left(\left[\frac{2n+2}{3}\right] + \left[\frac{2n+1}{3}\right]\right).$$

As an example, the logical circuit of the adder $C(3, 3)$ synthesized on the basis of formulas (1) and (2) in which $n = 3$ is presented in Fig. 1(a).

The logical circuit of the $C(3, 3)$ adder contains two OR elements and four exclusive OR elements with the thresholds one, two, four, and five. The complexity of the adder according to the number of inputs of logical elements is equal to $L(C(3, 3)) = 40$, whereas the processing speed determined by the depth of the scheme is 2τ, where τ is the averaged delay per one logical element.

5. DEVICE FOR ADDING AND SUBSTRACTION OF N NUMBERS

The method for the synthesis of the $C(n, 3)$ adder designed for the calculation of the operation $X_1 + X_2 + ... + X_n = S$ (mod 3) can be generalized to the case of the realization of the operation of the adding and subtraction of n numbers $\pm X_1 \pm X_2 \pm ... \pm X_n = S$ (mod 3) by a device interpreted as an $C^*(n, 3)$ adder.

It is known that, for the transformation of X_i into $-X_i$, it is necessary to fulfill the commutation of values of the low-order digit x_1^i and the high-order digit x_2^i of the number X_i; i.e., if $X_i = 2x_2^i + x_1^i$, then $-X_i = 2x_1^i + x_2^i$, where $i = 1, 2, ..., n$.

Table 1. Syntonization of the adder $C^*(3, 3)$

Syntonization signals						Realizable arithmetic operation
u_1	u_2	u_3	u_4	u_5	u_6	$\pm X_1 \pm X_2 \pm X_3 \pmod 3$
x_2^1	x_1^1	x_2^2	x_1^2	x_2^3	x_1^3	$X_1 + X_2 + X_3 \pmod 3$
x_2^1	x_1^1	x_2^2	x_1^2	x_1^3	x_2^3	$X_1 + X_2 - X_3 \pmod 3$
x_2^1	x_1^1	x_1^2	x_2^2	x_2^3	x_1^3	$X_1 - X_2 + X_3 \pmod 3$
x_2^1	x_1^1	x_1^2	x_2^2	x_1^3	x_2^3	$X_1 - X_2 - X_3 \pmod 3$
x_1^1	x_2^1	x_2^2	x_1^2	x_2^3	x_1^3	$-X_1 + X_2 + X_3 \pmod 3$
x_1^1	x_2^1	x_2^2	x_1^2	x_1^3	x_2^3	$-X_1 + X_2 - X_3 \pmod 3$
x_1^1	x_2^1	x_1^2	x_2^2	x_2^3	x_1^3	$-X_1 - X_2 + X_3 \pmod 3$
x_1^1	x_2^1	x_1^2	x_2^2	x_1^3	x_2^3	$-X_1 - X_2 - X_3 \pmod 3$

The structure of the logical circuit of the $C^*(n, 3)$ adder coincides with the structure of the $C(n, 3)$ adder scheme; i.e., $L(C^*(n, 3)) = L(C(n, 3))$.

The distinctive feature of the scheme of the $C^*(n, 3)$ adder consists in the fact that this scheme has $2n$ tuning inputs such that the syntonization signals $u_1, u_2, \ldots, u_n$ come to them. At that, $u_{2i-1}, u_{2i} \in \{x_1^i, x_2^i\}$, where $i = 1, 2, \ldots, n$. If an item X_i comes in the expression $\pm X_1 \pm X_2 \pm \ldots \pm X_n = S \pmod 3$ with the sign "plus," then $u_{2i-1} = x_2^i$ and $u_{2i} = x_1^i$; if it comes with the sign "minus," then $u_{2i-1} = x_1^i$ and $u_{2i} = x_2^i$.

The logical circuit of the $C^*(3, 3)$ device is presented in Fig. 1 (b); the syntonization table of the logical circuit $C^*(3, 3)$ for the realization of the operations $\pm X_1 \pm X_2 \pm \ldots \pm X_3 = S \pmod 3$ is shown in Table 1.

The $C^*(3, 3)$ logical circuit is oriented to the realization of any one of eight operations of the type $\pm X_1 \pm X_2 \pm X_3 = S \pmod 3$.

6. ADDER WITH CONTROL INPUT

Formulas (1) and (2) can be combined by means of the introduction of a binary parameter u as follows:

$$S(X^1, X^2, u) = \bigvee_{i=1}^{k_1} F_{3n+1}^{3i-1}(X^1, X^2, X^2, u), \tag{3}$$

where $k_1 = \left[\frac{2n+2}{3}\right]$. At that, $S(X^1, X^2, 1) = S_1(X^1, X^2)$ and $S(X^1, X^2, 0) = S_2(X^1, X^2)$.

Using formula (3), we can synthesize the logical circuit of an adder $C^{**}(n, 3)$ with control input. The logical circuit of the adder $C^{**}(n, 3)$ has $2n$ informational inputs such that the values of the variables x_1^1,

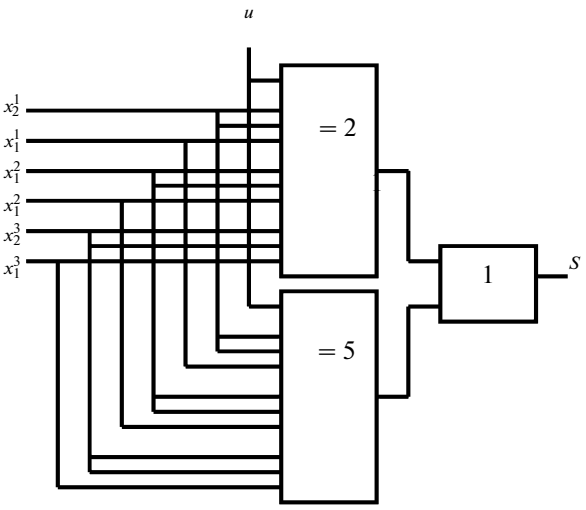

Fig. 2. Logical circuit of the adder $C^{**}(3, 3)$.

$x_2^1, x_1^2, x_2^2, \ldots, x_1^n, x_2^n$ come to them; it also has one control input to which the value of the control signal u is given, where $u \in \{0, 1\}$. On the unique output of the scheme $C^{**}(n, 3)$, the values of the functions S_1 or S_2 are realized (depending on the value of the control signal u).

The logical circuit of the $C^{**}(n, 3)$ adder contains $k_1 = \left[\frac{2n+2}{3}\right]$ EXCLUSIVE OR elements and one OR element, whereas its constructive complexity is

$$L(C^{**}(3, 3)) = (2n + 2) \cdot \left[\frac{2n+2}{3}\right].$$

The logical circuit of the $C^{**}(n, 3)$ adder is represented in Fig. 2 on the condition that $n = 3$. If $u = 1$, then the function of the low-order digit S_1 of the addition result is realized; if $u = 0$, then the function of the high-order digit S_2 of the addition result is realized. Table 2 is the truth table of the functions that are realized on the output of the adder $C^{**}(3, 3)$ depending on the value of the control signal u.

Table 2. Truth table of the adder $C^{**}(3, 3)$

Inputs						Outputs	
binary code of the first operand $X_1(x_2^1, x_1^1)$		binary code of the second operand $X_2(x_2^2, x_1^2)$		binary code of the third operand $X_3(x_2^3, x_1^3)$		addition result $S(X^1, X^2, u)$	
x_2^1	x_1^1	x_2^2	x_1^2	x_2^3	x_1^3	$S = S_2$, if $u = 0$	$S = S_1$, if $u = 1$
0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	1
0	0	0	0	1	0	1	0
0	0	0	1	0	0	0	1
0	0	0	1	0	1	1	0
0	0	0	1	1	0	0	0
0	0	1	0	0	0	1	0
0	0	1	0	0	1	0	0
0	0	1	0	1	0	0	1
0	1	0	0	0	0	0	1
0	1	0	0	0	1	1	0
0	1	0	0	1	0	0	0
0	1	0	1	0	0	1	0
0	1	0	1	0	1	0	0
0	1	0	1	1	0	0	1
0	1	1	0	0	0	0	0
0	1	1	0	0	1	0	1
0	1	1	0	1	0	1	0
1	0	0	0	0	0	1	0
1	0	0	0	0	1	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0
1	0	0	1	0	1	0	1
1	0	0	1	1	0	1	0
1	0	1	0	0	0	0	1
1	0	1	0	0	1	1	0
1	0	1	0	1	0	0	0

CONCLUSIONS

The method for the synthesis of n-operand adders modulo 3 given in the article allows us to synthesize logical circuites of adders that contain OR and exclusive OR elements with different thresholds. The main advantage of the schemes synthesized is their high operating speed, which is defined by their depths (number of levels). Moreover, the synthesized schemes of $C(n, 3)$, $C^*(n, 3)$, and $C^{**}(n, 3)$ adders have relatively small structural complexity, which is defined by the sum of the inputs of the logical elements constituting the corresponding schemes.

It should be noted that the computational devices that are synthesized using the method suggested in the paper are single-type. This, in turn, raises the uniformity of the modular structure as a whole and allows calculating using one and the same equipment for different values of the module [4].

The use of the described method for the synthesis of modular arithmetic devices allows us to obtain schemes that favorably differ from the analogues (by complexity and depth) obtained with the help of known CAD systems [10].

In the future, the suggested method for the synthesis of logical circuites of devices realizing the operations $X_1 + X_2 + ... + X_n = S \pmod 3$ and $\pm X_1 \pm X_2 \pm ... \pm X_n = S \pmod 3$ can be generalized for the case of the realization of addition and subtraction operations for n operands over an arbitrary modulo p.

REFERENCES

1. *Computers, Software, Engineering and Digital Devices*, Dorf, R.F., Eds., Taylor and Francis, 2006.
2. Chervyakov, N.I., Sakhnyuk, P.A., Shaposhnikov, A.V., and Ryadnov, S.A., *Modulyarnye parallel'nye vychislitel'nye struktury neiroprotsessornykh setei* (Modular Parallel Computational Structures of Neuroregulator Systems), Moscow: Fizmatlit, 2003.
3. Kornilov, A.I., Semenov, M.Yu., and Kalashnikov, V.S., Methods of Hardware-controlled Optimization of Adders for Two Operands in a System of Residual Classes, *Izv. Vysh. Uchebn. Zaved., Elektron.*, 2004, no. 1, pp. 75–82.
4. Avgul, L.B. and Kursenko, S.V., Syntesis of Adders for a Modified Termolecular System of Residual Classes on the Base of the Local Decoding Principle, *Avtom. Vychisl. Tekh.,* 1994, no. 4, pp. 3–12.
5. Suprun, V.P. and Gorodetskii, D.A., Method for Block-structural Synthesis of Modular Arithmetic Computational Devices, *Informatika*, 2009, no. 4(24), pp. 74–79.
6. Belarus Patent 3707, IPC G 06 F 7/49, Official Bull. no. 4(27), 2000, p. 210.
7. Belarus Patent 9600, IPC G 06 F 7/49, Official Bull. no. 4(57), p. 165.
8. Belarus Patent 10201, IPC G 06 F 7/48, 7/38, Official Bull. no. 1(60), pp. 152–153.
9. Belarus Patent 7000, IPC G 06 F 7/49, Official Bull. no. 2(45), p. 242.
10. Bibilo, P.N. and Gorodetskii, D.A., Automated Design of Modular Arithmetic Devices: Might CAD Replace an Engineer, *Avtom. Vychisl. Tekh.*, 2009, no. 2, pp. 15–27.