МАТЕМАТИЧНЕ ТА КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ

MATHEMATICAL AND COMPUTER MODELING

МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

UDC 681.142.01

METHODS FOR TABULAR IMPLEMENTATION OF ARITHMETIC OPERATIONS OF THE RESIDUES OF TWO NUMBERS REPRESENTED IN THE SYSTEM OF RESIDUAL CLASSES

Krasnobayev V. A. – Dr. Sc., Professor, Professor of Department of Electronics and Control Systems, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.

Yanko A. S. – PhD, Associate Professor, Associate Professor of the Department of Computer and Information Technologies and Systems, National University «Yuri Kondratyuk Poltava Polytechnic», Poltava, Ukraine.

Kovalchuk D. M. – Post-graduate student of Department of Electronics and Control Systems, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.

ABSTRACT

Context. Implementation of modular arithmetic operations of addition, subtraction and multiplication by a tabular method based on the use of the tabular multiplication code. The object of the study is the process of tabular implementation of basic arithmetic operations on the residues of numbers represented in the system of residual classes.

Objective. The goal of the work is to develop methods for the tabular implementation of the arithmetic operations of multiplication, addition and subtraction of the residues of two numbers based on the use of the tabular multiplication code.

Method. Tabular methods for implementing integer arithmetic modular operations of addition, subtraction and multiplication are proposed for consideration. In order to reduce the amount of equipment for a tabular operating unit of computer systems that implements modular operations of addition, subtraction and multiplication by reducing the coincidence circuits AND in the nodes of the tables for implementing arithmetic operations based on the code of table multiplication, two methods for performing arithmetic modular operations of addition and subtraction have been developed. These methods are based on the code of tabular multiplication, the use of which will reduce the amount of equipment of the tabular operating unit. Thus, despite the difference in the digital structure of the tables of modular operations of addition, subtraction and multiplication based on the use of the tabular multiplication code, two new tabular methods for implementing arithmetic modular operations of addition and subtraction have been created. Based on them, algorithms for tabular execution of modular arithmetic operations of addition and subtraction have been developed. Using these algorithms, it is possible to synthesize a structurally simple, highly reliable and fast table operating unit that operates in a system of residual classes, which is based on three separate permanent storage devices (read-only memory), each of which implements only one fourth of the corresponding complete table of values of the modular operation, what is earlier in the theory tabular arithmetic was supposed to be impossible.

Results. The developed methods are justified theoretically and studied when performing arithmetic modular operations of addition, subtraction and multiplication using tabular procedures.

Conclusions. The conducted examples of the implementation of integer arithmetic modular operations of addition and subtraction can be considered as presented experiments. The results obtained make it possible to recommend them for use in practice in the design of computer systems operating in a non-positional number system in residual classes. Prospects for further research may be to create a tabular method for implementing integer arithmetic modular division operations based on the use of the tabular multiplication code.

KEYWORDS: modular arithmetic operation, system of residual classes, tabular arithmetic, tabular multiplication code.

ABBREVIATIONS

CS is a computer system;
MCA is machine-computer arithmetic;
BPNS is a binary positional number system;
SRC is a system in residual classes;
TMC is a tabular multiplication code;
TOU is a tabular operating unit.

NOMENCLATURE

a_i is a residue an arbitrary modulo m_i of the number A represented in the SRC;

DOI 10.15588/1607-3274-2022-4-2

b_i is a residue an arbitrary modulo m_i of the number B represented in the SRC;

m_i is a SRC module;

M is a value of the informational numerical range;

γ_{a_i} (γ_{b_i}) is a sign of the TMC values of residues a_i and b_i accordingly;

γ_i is a generalized sign of the TMC;

γ_r is a sign of the TMC of the result of an arithmetic modular operation (addition or subtraction);

a_i' is a residue an arbitrary modulo m_i of the number A presented in the TMC;

b' is a residue an arbitrary modulo m_i of the number B presented in the TMC;

Φ (a_i', b_i') is a selection function by the values of the residues a_i' and b_i' the result of the operation of addition or subtraction in the corresponding nodes of the tables;

Φ_1 () is a dependence function of the implementation of the operation of modular addition, depending on the result of the operation of the operation of modular subtraction of two numbers in the SRC;

Φ_2 () is a dependence function of the implementation of the operation of modular subtraction, depending on the result of the operation of the operation of modular addition of two numbers in the SRC.

INTRODUCTION

It is known that one of the effective ways to increase the speed of a CS operating in the BPNS has led to the need to develop new MCA. MCA is based on the theory of residues of natural numbers and on the results of the proof of the Chinese remainder theorem. In the literature, such an MCA is called a non-positional number system in SRC [1–4]. Based on the properties of the SRC, its use allows to significantly increasing the speed of performing integer modular arithmetic operations of addition, subtraction and multiplication of the residues of numbers modulo SRC. In addition, such a property of the SRC as the low-bit capacity of the residues, the totality of which determines the non-positional code structure, makes it possible to effectively apply the tabular implementation of arithmetic operations [1, 2]. In the general case, the TOU of the CS for the implementation of arithmetic operations that are performed in a unitary code is a two-input ROM. For each of the ROM inputs, the number of input buses for the l-byte ($8l$ bits) CS is 2^{8l}. In this case, the total number of logic circuits AND in the ROM nodes (which basically determines the amount of equipment of the TOU of CS in the BPNS) is equal to the value $N_{l\ BPNS} = 2^{8l} \times 2^{8l} = 2^{16l}$. It is obvious that the table implementation of integer arithmetic operations in the usual BPNS is appropriate only for the value l=1 [5].

The search for ways to increase the efficiency of using tabular arithmetic necessitated the development and improvement of methods for tabular implementation of the main integer modular arithmetic operations: addition, subtraction and multiplication of the residues of numbers, aimed at reducing the number of ROM elements.

The object of study is the process of implementing arithmetic modular operations of addition, subtraction and multiplication in SRC.

The process of implementing arithmetic operations based on the BPNS involves the sequential processing of digits of numbers according to the rules determined by the content of this operation, and cannot be completed until the values of all intermediate results are sequentially determined, taking into account all connections between the digits. This drawback significantly affects the methods for implementing arithmetic operations and limits the speed of data processing. The number system in SRC has the valuable property of independence of the residues of the processed numbers in the accepted base system, which opens up wide opportunities in building not only new machine arithmetic, but also a fundamentally new circuit implementation of the data processing CS with the effective use of tabular methods.

The subject of study is the tabular methods for implementing arithmetic modular operations of addition, subtraction and multiplication.

The known tabular methods are distinguished by the complexity of implementation, low speed of implementation of basic arithmetic operations, as well as an increase in the amount of TOU equipment with an increase in the length of the bit grid, which is typical for the modern trend in the development of powerful computing systems.

The purpose of the work is to increase the efficiency of using tabular methods for performing basic arithmetic operations (addition, subtraction and multiplication) based on the representation of numbers in SRC and the use of a tabular multiplication code.

1 PROBLEM STATEMENT

In a formalized form, the statement of the problem of the article can be represented as a realization of two analytical relations. For the first method of performing a modular addition operation through the result of a modular subtraction operation, the first analytical relation is represented as: $(A+B) \bmod M = \Phi_1(\gamma_r \,\|\, \Phi(a_i',\ b_i')) = \Phi_1\left(m_i - \left\{[m_i - (\gamma_{a_i} \,\|\, a_i')] - (\gamma_{b_i} \,\|\, b_i')\right\}\right)$. For the second method of performing a modular subtraction operation through the result of a modular addition operation, the second analytical relation is represented as: $(A-B) \bmod M = \Phi_2(\gamma_r \,\|\, \Phi(a_i',\ b_i')) = \Phi_2\left\{(\gamma_{a_i} \,\|\, a_i') + \left[m_i - (\gamma_{b_i} \,\|\, b_i')\right]\right\}$. These methods implement the operations of modular addition and subtraction for the original two numbers A and B represented in the SRC as a set of residues $(a_1, a_2, ..., a_{i-1}, a_i, a_{i+1}, ..., a_n)$ and $(b_1, b_2, ..., b_{i-1}, b_i, b_{i+1}, ..., b_n)$ an arbitrary modulo m_i SRC, i.e. residues $a_i = A - \left[A/m_i\right] \cdot m_i$, $(i = \overline{1, n})$ the residues

DOI 10.15588/1607-3274-2022-4-2

are obtained from the successive division of the initial number by a set of mutually pairwise prime numbers m_i called bases or SRC modules. Such a representation of numbers (coding) makes it possible to construct data processing CSs in which the processing of all digits (residues) is performed in parallel in time.

Denote the generalized arithmetic operation as $\otimes$, which will be used as a sign (feature) of the arithmetic operation of modular multiplication, addition or subtraction. For numbers *A* and *B* represented by the SRC code The result of the operation can be represented as follows: $(A \otimes B) \bmod M = \left((a_1 \otimes b_1) \bmod m_1, \ldots, (a_n \otimes b_n) \bmod m_n\right)$. In this case, it is necessary that the following inequalities hold: $0 \le A < M$, $0 \le B < M$, $0 \le (A \otimes B) < M$, where $M = \prod_{i=1}^{n} m_i$. These requirements are due to the need to stay of the informational numerical range $[0, M)$ like numbers $A = (a_1, a_2, ..., a_n)$ and $B = (b_1, b_2, ..., b_n)$, as well as the result of the operation $(A \otimes B) \bmod M$.

2 REVIEW OF THE LITERATURE

The results of the search for ways to improve the performance of CS, effective methods for implementing the basic operations of the computational process and as a result of building high-speed and high-reliable systems, carried out over the past decades by various groups of researchers have confirmed in the opinion that within the BPNS one cannot expect significant satisfactory progress in these areas without a significant increasing the operating frequencies of the processor elements and the complication of the hardware part of the CS [1–12].

The results of research in the field of the creation of high-speed CS of well-known authors (Aksushskyi I.Ya. and Yuditskyi D.I. [1], Gregory R.T. and Krishnamurthy E.V. [6], Mohan P.V.A. [9] and others) showed that the use of SRC as a system of calculations of CS, intended for the implementation of integer arithmetic operations of addition, subtraction and multiplication numbers in the positive numerical range, significantly increases the speed of the solution of problems of a certain class [13].

In order to be able to build CS in SRC, it was necessary to develop fundamental methods for constructing MCA [1, 7, 14]. The implementation of the obtained theoretical and practical results will contribute to the creation, development and operation of real CS operating in a non-positional number system in the residual classes [15–20]. Prospects for further research may be to create a tabular method for implementing integer arithmetic modular division operations based on the use of the tabular multiplication code [16].

3 MATERIALS AND METHODS

Let's first consider the procedure for implementing the arithmetic operation of modular multiplication $(a_i \cdot b_i) \bmod m_i$ two residues a_i and b_i by an arbitrary modulo m_i respectively of the numbers *A* and *B* represented in the SRC. It is known that the table of values $(a_i \cdot b_i) \bmod m_i$ the result of the operation of modular multiplication is symmetrical with respect to the diagonals, verticals and horizontals passing (for m_i – an odd number) between the numbers $\frac{(m_i - 1)}{2}$ and $\frac{(m_i + 1)}{2}$.

Symmetry with respect to the left diagonal of the two-input table of the result of the operation is determined by the commutativity of the multiplication operation $a_i \cdot b_i = b_i \cdot a_i$. Symmetry with respect to the right diagonal of the table is determined by the fact that the condition $a_i \cdot b_i \equiv [(m_i - b_i) \cdot (m_i - a_i)] \bmod m_i$. Symmetry with respect to the vertical and horizontal of the table is determined from the condition of multiplicity modulo m_i of the sum of symmetric numbers $a_i \cdot b_i \equiv [m_i - a_i(m_i - b_i)] \bmod m_i$ as well as $a_i \cdot b_i \equiv [m_i - b_i(m_i - a_i)] \bmod m_i$.

Using the symmetry properties of the modular multiplication table of the residues of numbers, you can completely restore the complete table of values $a_i \cdot b_i (\bmod m_i)$ of the multiplication operation of residues using only 0.25 of its part. Hence, it becomes possible to simplify the table (reduce the number of two-input elements AND of the TOU corresponding to the nodes of the complete modular multiplication table. To solve the problem, it is necessary to introduce a sign (feature) that determines the location of the input residues of numbers in each of the four quadrants of the complete modular multiplication table. In [1] this sign is called tabular multiplication code.

Consider one of the possible options for encoding the input residues a_i and b_i tables of operation of modular multiplication modulo m_i by means of a specially introduced data compression code of the TMC. Values of the residues a_i (b_i) which is in the numerical range $\left[0, \frac{m_i - 1}{2}\right)$ can be encoded arbitrarily. Then the values of a_i (b_i) which is in the numerical range $\left[\frac{m_i + 1}{2}, m_i\right)$ encoded as the inverse of a number modulo $m_i - a_i$ or $m_i - b_i$. To distinguish the ranges of finding the values of the residues a_i and b_i the sign γ_{a_i} (γ_{b_i}) of the TMC is introduced defined as follows:

$$\gamma_{a_i}\ (\gamma_{b_i}) = \begin{cases} 0, \text{ if } 0 \le a_i(b_i) \le \dfrac{m_i - 1}{2}, \\ 1, \text{ if } \dfrac{m_i + 1}{2} \le a_i(b_i) \le m_i - 1. \end{cases} \quad (1)$$

DOI 10.15588/1607-3274-2022-4-2

The procedure for determining the result of a modular multiplication operation by means of TMC is as follows. If two residues are given modulo m_i of the form $a_i = (\gamma_{a_i} \| a_i'),\ b_i = (\gamma_{b_i} \| b_i'),$ where $\|$ – the mathematical sign of the concatenation operation (the operation of gluing, the operation of joining) and $0 \le a_i'(b_i') \le (m_i - 1)/2$. To get the product of these numbers modulo m_i, it is enough to get the product $a_i' \cdot b_i' (\mathrm{mod}\, m_i)$ and invert its generalized sign γ_i, in the event that γ_{a_i} is different from γ_{b_i}, i.e. the result of multiplying two residues modulo can be represented as $a_i \cdot b_i (\mathrm{mod}\, m_i) = (\gamma_i \| (a_i' \cdot b_i') \bmod m_i)$ on condition:

$$\gamma_i = \begin{cases} 0, \text{ if } \gamma_{a_i} = \gamma_{b_i}, \\ 1, \text{ if } \gamma_{a_i} \ne \gamma_{b_i}. \end{cases} \quad (2)$$

Until now, there are no methods for performing by means of TMC, arithmetic modular operations of adding and subtracting the residues of two numbers, which make it possible to reduce the number of two-input elements AND in the TOU CS. This hinders the further development of SRC in terms of the practical use of tabular arithmetic. The main difficulty lies in the fact that it is quite difficult to synthesize algorithms for performing modular operations due to the fact that tables $(a_i \otimes b_i) \bmod m_i$ performing modular operations of multiplication, addition and subtraction are different in their digital structure. When studying the digital properties of tables of modular addition and subtraction operations, the validity of expression (3) is shown:

$$\left[\left(\gamma_{a_i} \| a_i'\right) + \left(\gamma_{b_i} \| b_i'\right)\right] + \\ + \left\{\left[m_i - \left(\gamma_{a_i} \| a_i'\right)\right] - \left(\gamma_{b_i} \| b_i'\right)\right\} = 0 \left(\bmod m_i\right), \quad (3)$$

where $a_i = \left(\gamma_{a_i} \| a_i'\right),\ b_i = \left(\gamma_{b_i} \| b_i'\right)$ – are given residues modulo m_i presented in the TMC of the form $a_i = (\gamma_{a_i} \| a_i'),\ b_i = (\gamma_{b_i} \| b_i')$ with the condition $0 \le a_i'(b_i') \le (m_i - 1)/2$ for m_i odd number or $0 \le a_i'(b_i') \le m_i/2$ for m_i even number.

Expression (3) can be interpreted as an analytical dependence of the modular operations of addition and subtraction of the residues modulo numbers presented in the TMC.

It follows from expression (3) that in order to obtain the result of the modular addition operation by means of the TMC, it is sufficient to know the result of the modular subtraction and to obtain the result of the modular subtraction operation by means of the TMC, it is sufficient to know the result of the modular addition. That is, it becomes possible to effectively (from the point of view of reducing the ROM hardware) use the TMC not only to perform the modular multiplication operation, but also to implement the modular addition and subtraction operations.

Let's write expression (3) in the form (4). Expression (4) is the mathematical basis of the tabular method for performing the operation of modular addition using tables that implement the operation of modular subtraction:

$$\left(\gamma_{a_i} \| a_i'\right) + \left(\gamma_{b_i} \| b_i'\right) = m_i - \left\{\left[m_i - \left(\gamma_{a_i} \| a_i'\right)\right] - \left(\gamma_{b_i} \| b_i'\right)\right\}. \quad (4)$$

The method of performing the operation of modular addition using tables that implement the operation of modular subtraction can be represented as follows (Fig. 1).

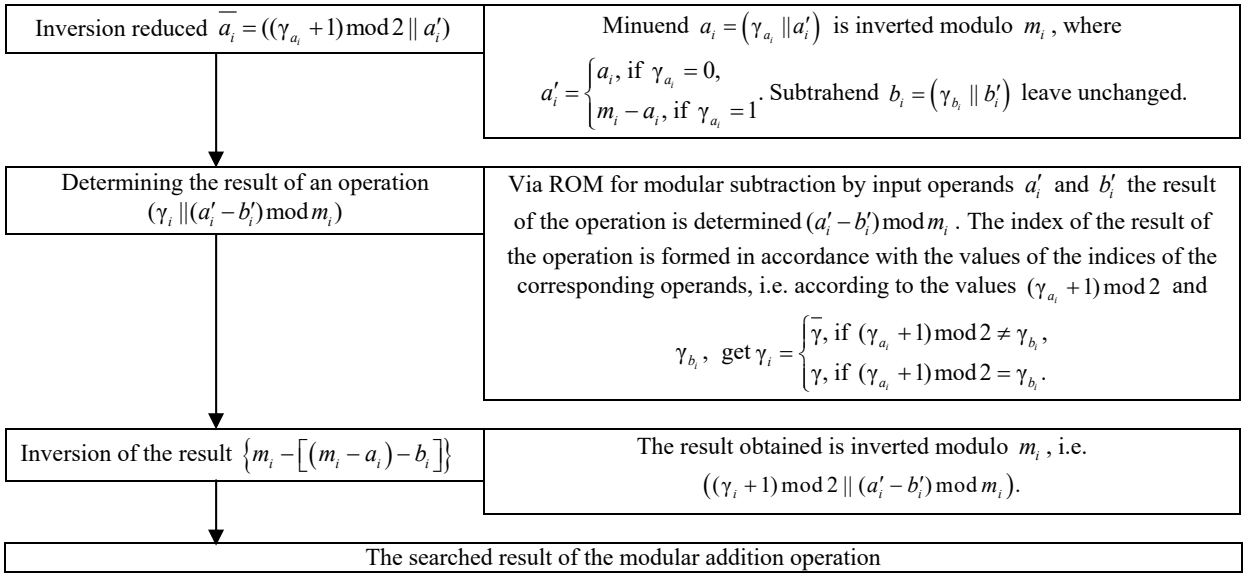

Figure 1 – The method of performing the operation of modular addition using tables that implement the operation of modular subtraction

DOI 10.15588/1607-3274-2022-4-2

The resulting method can be schematically represented in the form of 4 stages:

$$(a_i - b_i) \to \left[(m_i - a_i) - b_i\right] \to \left\{m_i - \left[(m_i - a_i) - b_i\right]\right\} \to$$
$$\to (a_i + b_i).$$

Let's write expression (3) in the form (5). Expression (5) is the mathematical basis of the tabular method for performing the operation of modular subtraction using tables that implement the operation of modular addition:

$$\left(\gamma_{a_i} \parallel a_i'\right) - \left(\gamma_{b_i} \parallel b_i'\right) = \left\{\left(\gamma_{a_i} \parallel a_i'\right) + \left[m_i - \left(\gamma_{b_i} \parallel b_i'\right)\right]\right\}, \quad (5)$$

i.e. it is possible to determine the result of the modular subtraction operation by means of a ROM that implements the modular addition operation.

Consider the result of a modular arithmetic operation in the TMC, which is represented as:

$$(\gamma_r \parallel \Phi(a_i', b_i')) \quad (6)$$

where γ_r – sign of the TMC of the result of a modular arithmetic operation.

Expression $\Phi(a_i', b_i')$ – a numeric value assigned to the table node of the corresponding arithmetic modular operation with coordinates a_i' and b_i'. If the residues are given $a_i = 2$ and $b_i = 4$ numbers modulo m_i =5, the expression $\Phi(a_i', b_i')$ is a numeric value assigned to the table node of the corresponding arithmetic modular operation with coordinates $a_i' = 2$ and $b_i' = 4$.

The method of performing the operation of modular subtraction using tables that implement the operation of modular addition can be represented as follows (Fig. 2).

Simplified schematically, the second method can be represented in the form of 3 stages:

$$(a_i + b_i) \to \left[a_i + (m_i - b_i)\right] \to (a_i - b_i).$$

When jointly implementing the arithmetic operations of addition and subtraction, the second method allows, in comparison with the first method, to implement the arithmetic modular subtraction operation in less time and with less hardware costs. Despite the difference in the digital structure of the tables of modular operations $(a_i \otimes b_i) \bmod m_i$ of addition, subtraction and multiplication (for example, for $m_i = 5,$ these are Tables 2–9), the developed first and second methods of addition-subtraction, which implement arithmetic modular operations, can reduce the number of elements of the TOU CS. This is achieved by simultaneously using only fourth part of each of the three complete addition, subtraction and multiplication tables by using TMC, which was previously thought to be impossible.

4 EXPERIMENTS

The experimental base of research is based on the theory of residues of natural numbers and on the results of the proof of the Chinese remainder theorem. The initial data in the form of bases (modules) m_i SRC are represented by a set of mutually pairwise prime numbers.

In the developing of the methods, the influence of the main properties of the SRC on the structure and principles of the functioning of the CS was used. Due to the low-bit capacity of the computational paths of the data processing CS presented in the SRC, there are possibilities for using (unlike the BPNS) tabular arithmetic, where the arithmetic operations of addition, subtraction and multiplication are performed almost in one clock cycle. The low-bit capacity of residuals in the representation of numbers in the SRC makes it possible to choose a wide range of options for system engineering solutions in the implementation of modular arithmetic operations tabular principle basis (based on the use of small ROM).

Tabular methods for the implementation of arithmetic modular operations of addition, subtraction and multiplication based on the use of the TMC are proposed for consideration. In the performing, using the tabular methods of modular arithmetic operations developed in the article, it was possible to reduce the amount of equipment of the TOU through which these operations are implemented. Note that with an increase in the length of the bit grid, which is typical for the modern trend in the development of powerful computing systems, the efficiency of using the proposed tabular methods for performing modular arithmetic operations increases significantly.

As experiments carried out in this article, we can consider a brief description of the structures and content of two methods (Fig. 1, 2) and two algorithms for the tabular implementation of modular arithmetic operations of addition-subtraction (Tables 10, 11).

The proposed methods are brought to algorithms, on the basis of which classes of patentable devices that implement these algorithms have been developed and for which Ukrainian patent have been received (state patent of Ukraine for the invention № 106343 from 11.08.2014. "A device for tabular implementation of arithmetic operations of multiplication and addition of numbers modulo m_i of the residual class.").

Some of the results obtained in the article are a definite contribution to the theory and practice of tabular arithmetic, which can be used to create CS in the SRC.

5 RESULTS

As a demonstration of the effectiveness of the developed methods, consider examples of a specific implementation of arithmetic modular operations of multiplication, addition and subtraction for a module equal to the value $m_i = 5.$ In this case, for the tabular method of the table for the implementation of modular operations of using TMC the initial data are presented in Tables 1–9. The first and second algorithms for performing modular addition and subtraction operations, respectively, by the first and second methods, are presented in Table 10 and Table 11, respectively.

DOI 10.15588/1607-3274-2022-4-2

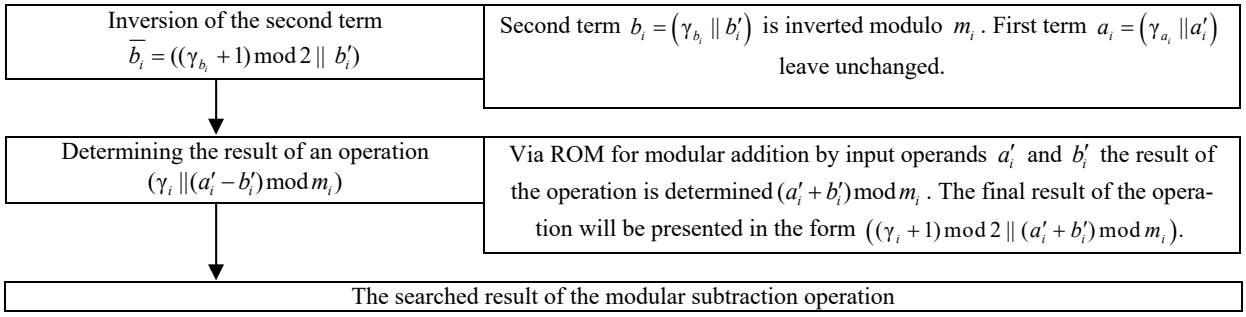

Figure 2 – The method of performing the operation of modular subtraction using tables that implement the operation of modular addition

Table 1 – Table multiplication code

a_i	TMC		a_i	TMC	
	γ_{a_i}	a_i'		γ_{a_i}	a_i'
1	0	1	3	1	2
2	0	2	4	1	1

Table 2 – Full table of modular multiplication

b_i \ a_i	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Table 3 – Full table of modular addition

b_i \ a_i	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Table 4 – Full table of modular subtraction

b_i \ a_i	0	1	2	3	4
0	0	1	2	3	4
1	4	0	1	2	3
2	3	4	0	1	2
3	2	3	4	0	1
4	1	2	3	4	0

Table 5 – First table of modular multiplication

b_i \ a_i		1	2
		4	3
1	4	1	2
2	3	2	1

Table 6 – First table of modular subtraction

b_i \ a_i		1	2
		4	3
1	4	0	1
2	3	1	0

Table 7 – Second table of modular subtraction

b_i \ a_i		2	1
		3	4
1	4	2	2
2	3	1	2

Table 8 – First table of modular addition

b_i \ a_i		1	2
		4	3
1	4	2	2
2	3	2	1

DOI 10.15588/1607-3274-2022-4-2

Table 9 – Second table of modular addition

b_i \ a_i		2	1
		3	4
1	4	1	0
2	3	0	1

In the implementing modular operations using the first algorithm, the TOU CS is based on three ROMs. The first ROM implements the II-quadrant of the complete multiplication table (Table 5); the second and third ROMs implement, respectively, I (Table 7) and II (Table 6) quadrants of the complete subtraction table).

In the implementing modular operations using the second algorithm, the TOU CS is also based on three ROMs, each of which implements 0.25 parts of the corresponding complete table of modular multiplication operations (Table 2) and addition (Table 3). The first ROM implements the II-quadrant of the complete multiplication table (Table 5); the second and third ROMs implement, respectively, I (Table 9) and II (Table 8) quadrants of the complete addition table. In this regard, the TMC acquired a new quality and became a universal tabular code for performing three arithmetic modular operations: addition, subtraction and multiplication.

In the implementing the arithmetic operations of addition and subtraction, the second proposed universal method makes it possible to implement a modular operation in less time and with less hardware costs (compared to the first method).

Table 10 – Algorithm for executing the first method

1	2	3	4	5
$(a_i - b_i) \bmod m_i$	$\gamma_{a_i} = \gamma_{b_i}$ $(\gamma_i = 0)$	Quadrant II is used (Table 6) of the complete table of modular subtraction	$\gamma_{a_i} = \gamma_{b_i} = 0$	The result is determined directly by the values of the nodes in Table 6 ($\gamma_r = \gamma_i$)
			$\gamma_{a_i} = \gamma_{b_i} = 1$	The result is determined by inverting modulo m_i the node values of Table 6 ($\gamma_r = (\gamma_i + 1) \bmod 2$)
	$\gamma_{a_i} \neq \gamma_{b_i}$ $(\gamma_i = 1)$	Quadrant I is used (Table 7) of the complete table of modular subtraction	$\gamma_{a_i} = 1,\ \gamma_{b_i} = 0$	The result is determined directly by the values of the nodes in Table 7 ($\gamma_r = \gamma_i$)
			$\gamma_{a_i} = 0,\ \gamma_{b_i} = 1$	The result is determined directly by the values of the nodes in Table 7 ($\gamma_r = \gamma_i$)
$(a_i + b_i) \bmod m_i$	$\gamma_{a_i} = \gamma_{b_i}$ $(\gamma_i = 0)$	Quadrant I is used (Table 7) of the complete table of modular subtraction	$\gamma_{a_i} = \gamma_{b_i} = 0$	The result is determined directly by the values of the nodes in Table 7 ($\gamma_r = \gamma_i$)
			$\gamma_{a_i} = \gamma_{b_i} = 1$	The result is determined by inverting modulo m_i the node values of Table 7 ($\gamma_r = (\gamma_i + 1) \bmod 2$)
	$\gamma_{a_i} \neq \gamma_{b_i}$ $(\gamma_i = 1)$	Quadrant II is used (Table 6) of the complete table of modular subtraction	$\gamma_{a_i} = 1,\ \gamma_{b_i} = 0$	The result is determined directly by the values of the nodes in Table 6 ($\gamma_r = \gamma_i$)
			$\gamma_{a_i} = 0,\ \gamma_{b_i} = 1$	The result is determined by inverting modulo m_i the node values of Table 6 ($\gamma_r = (\gamma_i + 1) \bmod 2$)

Table 11 – Algorithm for executing the second method

1	2	3	4	5
$(a_i - b_i) \bmod m_i$	$\gamma_{a_i} = \gamma_{b_i}$ $(\gamma_i = 0)$	Quadrant I is used (Table 9) of the complete table of modular addition	$\gamma_{a_i} = \gamma_{b_i} = 0$	The result is determined by inverting modulo m_i the node values of Table 9 ($\gamma_r = (\gamma_i + 1) \bmod 2$)
			$\gamma_{a_i} = \gamma_{b_i} = 1$	The result is determined directly by the values of the nodes in Table 9 ($\gamma_r = \gamma_i$)
	$\gamma_{a_i} \neq \gamma_{b_i}$ $(\gamma_i = 1)$	Quadrant II is used (Table 8) of the complete table of modular addition	$\gamma_{a_i} = 1,\ \gamma_{b_i} = 0$	The result is determined by inverting modulo m_i the node values of Table 8 ($\gamma_r = (\gamma_i + 1) \bmod 2$)
			$\gamma_{a_i} = 0,\ \gamma_{b_i} = 1$	The result is determined directly by the values of the nodes in Table 8 ($\gamma_r = \gamma_i$)
$(a_i + b_i) \bmod m_i$	$\gamma_{a_i} = \gamma_{b_i}$ $(\gamma_i = 0)$	Quadrant II is used (Table 8) of the complete table of modular addition	$\gamma_{a_i} = \gamma_{b_i} = 0$	The result is determined directly by the values of the nodes in Table 8 ($\gamma_r = \gamma_i$)
			$\gamma_{a_i} = \gamma_{b_i} = 1$	The result is determined by inverting modulo m_i the node values of Table 8 ($\gamma_r = (\gamma_i + 1) \bmod 2$)
	$\gamma_{a_i} \neq \gamma_{b_i}$ $(\gamma_i = 1)$	Quadrant I is used (Table 9) of the complete table of modular addition	$\gamma_{a_i} = 1,\ \gamma_{b_i} = 0$	The result is determined directly by the values of the nodes in Table 9 ($\gamma_r = \gamma_i$)
			$\gamma_{a_i} = 0,\ \gamma_{b_i} = 1$	The result is determined by inverting modulo m_i the node values of Table 9 ($\gamma_r = (\gamma_i + 1) \bmod 2$)

DOI 10.15588/1607-3274-2022-4-2

To confirm the efficiency of the developed methods and algorithms, consider some examples for the residues are given $a_i = 2$ and $b_i = 4$ numbers modulo m_i=5, which in the TMC are presented in the form $a_i = (\gamma_{a_i} \| a_i') = (0 \| 2)$ and $b_i = (\gamma_{b_i} \| b_i') = (1 \| 1)$, look at expression (1) and Table 1. In accordance with expression (2), get that $\gamma_i = (\gamma_{a_i} + \gamma_{b_i}) \bmod 2$.

Consider performing the operations of multiplying the residues $a_i = 2$ and $b_i = 4$ numbers modulo m_i =5 using TMC. Using expression (6), the result of the operation will be represented as $(a_i \cdot b_i) \bmod m_i = (\gamma_r \| \Phi(a_i', b_i'))$. In accordance with expression (2) get, $\gamma_i = 1$, $\gamma_r = \gamma_i = 1$, and $\Phi(a_i', b_i') = 2$ (Table 5). Thus, the result of modular arithmetic multiplication $(a_i \cdot b_i) \bmod m_i = (2 \cdot 4) \bmod 5 = (\gamma_r \| \Phi(a_i', b_i')) = (1 \| 2) = 3$. Check: $(2 \cdot 4) = 3 (\bmod 5)$, see Table 1, 2 and 5.

Consider the execution of the modular addition operation $(a_i + b_i) \bmod m_i$ for the first and second algorithms (the first and second methods):

The first algorithm (Table 10). The result of the modular addition is presented in the form $(\gamma_r \| \Phi(a_i', b_i'))$, where $\gamma_r = (\gamma_i + 1) \bmod 2 = (1+1) = 0 (\bmod 2)$ and $\Phi(a_i', b_i') = 1$ (Table 6). Check: $(2+4) = 1 (\bmod 5)$, see Table 1, 3, 6 and 10.

The second algorithm (Table 11). The result of the modular addition operation is represented in the form $(\gamma_r \| \Phi(a_i', b_i'))$, where $\gamma_r = (\gamma_i + 1) \bmod 2 = (1+1) \bmod 2 = 0$. By values of $a_i' = 2$ and $b_i' = 1$, in Table 9, the value is 1. The result of the operation will be as follows $(\gamma_r \| \Phi(a_i', b_i')) = (0 \| 1)$. Check: $(2+4) = 1 (\bmod 5)$, see Table 1, 3, 9 and 11.

Consider the operation of modular subtraction for the first and second algorithms (the first and second methods):

The first algorithm (Table 10). The result of the modular subtraction operation is represented as $(\gamma_r \| \Phi(a_i', b_i'))$, where $\Phi(a_i', b_i')$ – the value assigned to the node in the second modular subtraction table (Table 7) with coordinates $a_i' = 2$ and $b_i' = 1$. In accordance with the first algorithm (Table 10), given that $\gamma_i = 1$, sign value of γ_r the result of the modular subtraction operation is $\gamma_r = \gamma_i = 1$. The value of $\Phi(a_i', b_i')$ assigned to the node of Table 7 with coordinates $a_i' = 2$ and $b_i' = 1$, equals $\Phi(a_i' = 2, b_i' = 1) = 2$. The result of the operation of modular subtraction $(a_i - b_i)$ in the TMC, in accordance with the first algorithm, will be equal to $(\gamma_r \| \Phi(a_i', b_i')) = (1 \| 2)$. Check: $(2-4) \bmod 5 = 3 (\bmod 5)$, see Table 1, 4, 7 and 10.

The second algorithm (Table 11). Consider the implementation of the modular subtraction operation $(a_i - b_i)$ in the TMC, the result of the operation will be presented in the form $(\gamma_r \| \Phi(a_i', b_i'))$, where $\gamma_r = \gamma_i = 1$ $\Phi(a_i' = 2, b_i' = 1) = 2$ (Table 8). The result of the operation of modular subtraction in the TMC, in accordance with the second algorithm, will be equal to $(\gamma_r \| \Phi(a_i', b_i')) = (1 \| 2)$. Check: $(2-4) \bmod 5 = 3 (\bmod 5)$, see Table 1, 4, 8 and 11.

The results of the experimental or theoretical data obtained in the article consist in the development of two methods and two algorithms for the implementation of modular arithmetic operations of addition-subtraction in the SRC, by means of the TMC. The obtained scientific and theoretical results of the article represent a significant step forward compared to previous studies in the field of the theory of the implementation of modular operations using the tabular principle. So, Fig. 1 shows a method for performing a modular addition operation using table data that implements a modular subtraction operation, and Fig. 2 shows a method for performing a modular subtraction operation using table data that implements a modular addition operation. For this, digital data of tables for the implementation of integer modular addition-subtraction operations were developed (Tables 1–9). When developing the theoretical part, the methods for performing arithmetic modular operations presented in the article, various methods of cognition were used. For example, when synthesizing a non-positional code structure in the SRC, the induction method was used, and when implementing modular operations in the SRC, the deduction method and others were used. Based on the developed methods, algorithms for performing modular arithmetic operations of addition-subtraction in the SRC are presented (Tables 10, 11).

6 DISCUSSION

The analysis and assessment of the reliability of the results is based, firstly, on the correct use of the rules of tabular arithmetic. Secondly, on a clear and complete use of the properties of the SRC. And, finally, thirdly, the reliability of the results is confirmed by examples of tabular implementation of arithmetic modular addition-subtraction operations in the SRC, by means of the TMC, for specific values of the residues of numbers in the SRC. The scientific results obtained in this article refute similar results of prominent scientists in the field of implementation of tabular modular operations. So, for example, in chapter 6 “Computer components in the system of residual classes”, 6.3 “Fundamentals of tabular arithmetic”, on page 337 in the monograph of Aksushskyi I.Ya. and Yuditskyi D.I. “Machine arithmetic in residual classes”, who are the founders of scientific and technical developments in the field of the SRC in the USSR, the following is noted “... the tabular multiplication code is fundamentally unsuitable for use in the addition operation ...”[1]. This conclusion contradicts the conclusion of this article about the effective use of the TMC not only for the operation of modular multiplication, but also for the implementation of modular addition-subtraction operations. The

DOI 10.15588/1607-3274-2022-4-2

practical application of the results of the article is possible when creating a tabular operating device of a computer system operating in the SRC. The expediency of further research in the field of application of tabular arithmetic is due to the fact that this approach makes it possible to create high-speed and reliable computer systems.

CONCLUSIONS

A scientific and technical problem has been solved, which consists in developing methods and algorithms for the tabular implementation of the arithmetic operations of multiplication, addition and subtraction of the residues of two numbers based on the use of the tabular multiplication code.

The scientific novelty of obtained results is that two new tabular methods for the implementation of arithmetic modular operations of addition and subtraction have been created based on the use of the TMC, despite the difference in the digital structure of the tables of these modular operations. The use of the developed methods makes it possible to reduce the number of TOU CS equipment that implements the modular operations of addition, subtraction and multiplication by reducing the matching circuits AND in the nodes of the tables for implementing arithmetic operations based on the TMC. Based on these methods, algorithms for the tabular implementation of modular arithmetic operations of addition and subtraction have been developed. With the help of these algorithms, it is possible to synthesize a structurally simple, high-reliable and high-speed TOU CS operating in the SRC, which is based on three separate ROMs, each of which implements only 0.25 of the corresponding complete table of values of the modular operation, which was previously assumed impossible in the theory of tabular arithmetic.

The practical significance of obtained results is that in the performing, using the tabular methods developed in the article, modular arithmetic operations, it was possible to reduce 75% of the equipment of the TOU CS, through which these operations are implemented. This, in turn, as shown by the calculations, depending on the length of the bit grid of the CS, made it possible to reduce to $\approx$ (50–60)% of the equipment of the TOU CS in the SRC. Note that with an increase in the bit grid length CS, which is typical for the modern trend in the development of powerful computing systems, the efficiency of using the proposed tabular methods for performing modular arithmetic operations increases significantly. Some of the results obtained in the article are a definite contribution to the theory and practice of tabular arithmetic and it can be used when creating a CS in the SRC.

Prospects for further research are to create a tabular method for implementing integer arithmetic modular division operations based on the use of the TMC.

REFERENCES

1. Akushskij I. Ya., Yudickij D. I. Mashinnaya arifmetika v ostatochnyx klassax. Moscow , Sov. radio, 1968, 440 p.
2. Gérard B., Kammerer J.-G., Merkiche N. Contributions to the Design of Residue Number System Architectures, *Proceedings 22nd IEEE International Symposium on Computer Arithmetic*. France, 2015, pp. 105–112. DOI: 10.1109/ARITH.2015.25.
3. Bayoumi M., Jullien G. and Miller W. A VLSI implementation of residue adders, *IEEE Transactions on Circuits and Systems,* March 1987, Vol. 34, № 3, pp. 284–288. DOI: 10.1109/TCS.1987.1086130
4. Chao Huang D. Peterson, Rauch H., Teague J. and Fraser D. Implementation of a fast digital processor using the residue number system, *IEEE Transactions on Circuits and Systems,* January 1981, Vol. 28, No. 1, pp. 32–38. DOI: 10.1109/TCS.1981.1084905.
5. Ulman Z., Czyzak M. and Zurada J. Effective RNS scaling algorithm with the Chinese remainder theorem decomposition, *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing,* 1993, Vol. 2, pp. 528–531 DOI: 10.1109/PACRIM.1993.407305.
6. Gregory R. T., Krishnamurthy E. V. Residue or Modular Arithmetic. In: Methods and Applications of Error-Free Computation, *Texts and Monographs in Computer Science.* New York, Springer, 1984. DOI: https://doi.org/10.1007/978-1-4612-5242-9_1
7. Safari A., Nugent J. and Kong Y. Novel implementation of full adder based scaling in Residue Number Systems, *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2013, pp. 657–660, DOI: 10.1109/MWSCAS.2013.6674734.
8. Wei S. Fast signed-digit arithmetic circuits for residue number systems, *2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS*), December, 2015, pp. 344–347. DOI: 10.1109/ICECS.2015.7440319.
9. Mohan P. V. A. Residue Number Systems: Theory and Applications. Birkhäuser Basel, Springer International Publishing, Switzerland, 2016, 351 p. DOI: 10.1007/978-3-319-41385-3_2 9
10. Krasnobayev V., Kuznetsov A., Yanko A., Akhmetov B., Kuznetsova T. Processing of the residuals of numbers in real and complex numerical domains, *Lecture Notes on Data Engineering and Communications Technologies*. Berlin, Springer, 2021, pp. 529–555. DOI: 10.1007/978-3-030-43070-2_24
11. Piestrak S. J. Design of residue generators and multioperand modular adders using carry-save adders, *Proceedings 10th IEEE Symposium on Computer Arithmetic, 26–28 June,* 1991, pp. 100–107. DOI: 10.1109/ARITH.1991.145540
12. Gorbenko I., Hanzia R. Examination and implementation of the fast method for computing the order of elliptic curve, *Eastern-European Journal of Enterprise Technologies*, 2017, Vol. 2, No. 9(86), pp. 11–21. DOI: 10.15587/1729-4061.2017.95194
13. Krasnobayev V., Kuznetsov A., Yanko A., Koshman S., Zamula A. and Kuznetsova T. Data processing in the system of residual classes : monograph, ASC Academic Publishing, 2019, 208 p. ISBN: 978-0-9989826-6-3 (Hardback), ISBN: 978-0-9989826-7-0 (Ebook).
14. Szabo N. S., Tanaka R. I. Residue Arithmetic and Its Applications to Computer Technology. New York, McGraw-Hill, 1967, 236 p.
15. Kazymyrov O., Oliynykov R., Raddum H. Influence of addition modulo 2n on algebraic attacks, *Cryptography and Communications*, April 2016, Vol. 8, Issue 2, pp. 277–289. DOI: https://doi.org/10.1007/s12095-015-0136-7
16. Krasnobayev V. A., Yanko A. S., Koshman S. A. Algorithms of data processing in the residual classes system, *2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*. Kharkov, 2017, pp. 117–121. DOI: 10.1109/INFOCOMMST.2017.8246363
17. Arnold M. G. The Residue Logarithmic Number System: Theory and Implementation, *17th IEEE International Symposium on Computer Arithmetic*, 2005, pp. 196–205 DOI: 10.1109/ARITH.2005.44

DOI 10.15588/1607-3274-2022-4-2

18. Gavrylenko S. Y., Chelak V. V., Semenov S. G. Development of method for identification the computer system state based on the decision tree with multi-dimensional nodes, *Radio Electronics, Computer Science, Control*, 2022, No. (2), pp. 113–121. DOI: https://doi.org/10.15588/1607-3274-2022-2-11
19. Wang Y., Song X., Aboulhamid M., Shen H. Adder based residue to binary number converters for (2/sup n/–1, 2/sup n/, 2/sup n/+1), *IEEE Transactions on Signal Processing,* July 2002, Vol. 50, № 7, pp. 1772–1779. DOI: 10.1109/TSP.2002.1011216
20. Wang Wei, Swamy M. N. S., Ahmad M. O., Wang Yuke A study of the residue-to-binary converters for the three-moduli sets, *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, Febuary* 2003, Vol. 50, No. 2, pp. 235–243. DOI: 10.1109/TCSI.2002.808191

Received 09.09.2022.
Accepted 26.10.2022.

УДК 681.142.01

МЕТОДИ ТАБЛИЧНОЇ РЕАЛІЗАЦІЇ АРИФМЕТИЧНИХ ОПЕРАЦІЙ ЗАЛИШКІВ ДВОХ ЧИСЕЛ, ПРЕДСТАВЛЕНИХ У СИСТЕМІ ЗАЛИШКОВИХ КЛАСІВ

Краснобаєв В. А. – д-р техн. наук, професор, професор кафедри електроніки та управляючих систем Харківського національного університету імені В. Н. Каразіна, Харків, Україна.

Янко А. С. – канд. техн. наук, доцент, доцент кафедри комп'ютерних та інформаційних технологій і систем Національного університету «Полтавська політехніка імені Юрія Кондратюка», Полтава, Україна.

Ковальчук Д. М. – аспірант кафедри електроніки та управляючих систем Харківського національного університету імені В. Н. Каразіна, Харків, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто задачу реалізації модульних арифметичних операцій додавання, віднімання та множення табличним методом на основі використання коду табличного множення. Об'єктом дослідження є процес реалізації арифметичних модульних операцій додавання, віднімання та множення. Мета роботи – розробити методи табличної реалізації арифметичних операцій додавання, віднімання та множення залишків двох чисел на основі використання коду табличного множення.

Метод. Запропоновано до розгляду табличні методи реалізації цілочисельних арифметичних модульних операцій додавання, віднімання та множення. З метою скорочення кількості обладнання табличного операційного пристрою комп'ютерних систем, що реалізує модульні операції додавання, віднімання та множення, за рахунок скорочення схем збігу І у вузлах таблиць реалізації арифметичних операцій, на основі коду табличного множення, розроблено два методи виконання арифметичних модульних операцій додавання та віднімання. В основу даних методів покладено код табличного множення, використання якого дозволить зменшити кількість обладнання табличного операційного пристрою. Таким чином, незважаючи на відмінність цифрової структури таблиць модульних операцій додавання, віднімання та множення, на основі використання коду табличного множення, створено два нові табличні методи реалізації арифметичних модульних операцій додавання та віднімання. На їх основі розроблено алгоритми табличного виконання модульних арифметичних операцій додавання та віднімання. За допомогою цих алгоритмів можна синтезувати конструктивно простий, високонадійний та швидкодіючий табличний операційний пристрій, що функціонує в системі залишкових класів, основу якого складають три окремих постійних запам'ятовуючих пристрої, кожен з яких реалізує лише одну четверту частину відповідної повної таблиці значень модульної операції, що раніше в теорії табличної арифметики передбачалося неможливим.

Результати. Розроблені методи обґрунтовані теоретично та досліджені при виконанні арифметичних модульних операцій додавання, віднімання та множення за допомогою табличних процедур.

Висновки. Проведені приклади реалізації цілочисельних арифметичних модульних операцій додавання та віднімання можна розглядати як представлені експерименти. Отримані результати дозволяють рекомендувати їх використання на практиці проектування комп'ютерних систем, що функціонують у непозиційній системі числення в залишкових класах. Перспективи подальших досліджень можуть полягати у створенні табличного методу реалізації цілочисельної арифметичної модульної операцій ділення на основі використання коду табличного множення.

КЛЮЧОВІ СЛОВА: модульна арифметична операція, система залишкових класів, таблична арифметика, код табличного множення.

УДК 681.142.01

МЕТОДЫ ТАБЛИЧНОЙ РЕАЛИЗАЦИИ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ ОСТАТКОВ ДВУХ ЧИСЕЛ, ПРЕДСТАВЛЕННЫХ В СИСТЕМЕ ОСТАТОЧНЫХ КЛАССОВ

Краснобаев В. А. – д-р техн. наук, профессор, профессор кафедры электроники и управляющих систем Харьковского национального университета имени В. Н. Каразина, Харьков, Украина.

Янко А. С. – канд. техн. наук, доцент, доцент кафедры компьютерных и информационных технологий и систем Национального университета «Полтавская политехника имени Юрия Кондратюка», Полтава, Украина.

Ковальчук Д. Н. – аспирант кафедры электроники и управляющих систем Харьковского национального университета имени В. Н. Каразина, Харьков, Украина.

АННОТАЦИЯ

Актуальность. Рассмотрена задача реализация модульных арифметических операций сложения, вычитания и умножения табличным методом на основе использования кода табличного умножения. Объектом исследования являлась процесс реализации арифметических модульных операций сложения, вычитания и умножения. Цель работы – разработать методы табличной реализации арифметических сложения, вычитания и умножения остатков двух чисел на основе использования кода табличного умножения.

Метод. Предложены к рассмотрению табличные методы реализации целочисленных арифметических модульных операций сложения, вычитания и умножения. С целью сокращения количества оборудования табличного операционного устройства компьютерных систем, реализующего модульные операции сложения, вычитания и умножения, за счет сокращения

DOI 10.15588/1607-3274-2022-4-2

схем совпадения И в узлах таблиц реализации арифметических операций, на основе кода табличного умножения, разработаны два метода выполнения арифметических модульных операций сложения и вычитания. В основу данных методов положен код табличного умножения, использования которого позволит уменьшить количество оборудования табличного операционного устройства. Таким образом, несмотря на различие цифровой структуры таблиц модульных операций сложения, вычитания и умножения, на основе использования кода табличного умножения, созданы два новых табличных методов реализации арифметических модульных операций сложения и вычитания. На их основе разработаны алгоритмы табличного выполнения модульных арифметических операций сложения и вычитания. С помощью этих алгоритмов можно синтезировать конструктивно простое, высоконадежное и быстродействующее табличное операционное устройство, функционирующая в системе остаточных классов, основу которого составляют три отдельных постоянных запоминающих устройства, каждый из которых реализует только одну четвертую часть соответствующей полной таблицы значений модульной операции, что ранее в теории табличной арифметики предполагалось невозможным.

Результаты. Разработанные методы обоснованы теоретически и исследованы при выполнении арифметических модульных операций сложения, вычитания и умножения с помощью табличных процедур.

Выводы. Проведенные примеры реализации целочисленных арифметических модульных операций сложения и вычитания можно рассматривать в качестве представленных экспериментов. Полученные результаты позволяют рекомендовать их для использования на практике проектирования компьютерных систем, функционирующих в непозиционной системе счисления в остаточных классах. Перспективы дальнейших исследований могут заключаться в создании табличного метода реализации целочисленной арифметической модульной операций деления на основе использования кода табличного умножения.

КЛЮЧЕВЫЕ СЛОВА: модульная арифметическая операция, система остаточных классов, табличная арифметика, табличный код умножения.

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Акушский И. Я. Машинная арифметика в остаточных классах / И. Я. Акушский, Д. И. Юдицкий. – Москва : Сов. радио, 1968. – 440 с.
2. Gérard B. Contributions to the Design of Residue Number System Architectures / B. Gérard, J.-G. Kammerer, N. Merkiche // Proceedings 22nd IEEE International Symposium on Computer Arithmetic. – France, 2015. – P. 105–112. DOI: 10.1109/ARITH.2015.25.
3. Bayoumi M. A VLSI implementation of residue adders / M. Bayoumi, G. Jullien and W. Miller // IEEE Transactions on Circuits and Systems. – March 1987. – Vol. 34, № 3. – P. 284–288. DOI: 10.1109/TCS.1987.1086130
4. Implementation of a fast digital processor using the residue number system / [Chao Huang D. Peterson, H. Rauch, J. Teague and D. Fraser] // IEEE Transactions on Circuits and Systems. – January 1981. – Vol. 28, № 1. – P. 32–38. DOI: 10.1109/TCS.1981.1084905.
5. Ulman Z. Effective RNS scaling algorithm with the Chinese remainder theorem decomposition / Z. Ulman, M. Czyzak and J. Zurada // Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing. – 1993. – Vol. 2. – P. 528–531 DOI: 10.1109/PACRIM.1993.407305.
6. Gregory R. T. Residue or Modular Arithmetic. In: Methods and Applications of Error-Free Computation / R. T. Gregory, E. V. Krishnamurthy // Texts and Monographs in Computer Science. – New York : Springer, 1984. DOI: https://doi.org/10.1007/978-1-4612-5242-9_1
7. Safari A. Novel implementation of full adder based scaling in Residue Number Systems / A. Safari, J. Nugent and Y. Kong, // 2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS). – 2013. – P. 657–660. DOI: 10.1109/MWSCAS.2013.6674734.
8. Wei S. Fast signed-digit arithmetic circuits for residue number systems / S. Wei // 2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS). – December, 2015. – P. 344–347. DOI: 10.1109/ICECS.2015.7440319.
9. Mohan P. V. A. Residue Number Systems: Theory and Applications / P. V. A. Mohan. – Birkhäuser Basel: Springer International Publishing, Switzerland, 2016. – 351 p. DOI: 10.1007/978-3-319-41385-3_2 9
10. Krasnobayev V. Processing of the residuals of numbers in real and complex numerical domains / [V. Krasnobayev, A. Kuznetsov, A. Yanko et al.] // Lecture Notes on Data Engineering and Communications Technologies. – Berlin : Springer, 2021. – P. 529–555. DOI: 10.1007/978-3-030-43070-2_24
11. Piestrak S. J. Design of residue generators and multioperand modular adders using carry-save adders / S. J. Piestrak // Proceedings 10th IEEE Symposium on Computer Arithmetic, 26–28 June, 1991. – P. 100–107. DOI: 10.1109/ARITH.1991.145540
12. Gorbenko I. Examination and implementation of the fast method for computing the order of elliptic curve / I. Gorbenko, R. Hanzia // Eastern-European Journal of Enterprise Technologies. – 2017. – Vol. 2, № 9(86). – P. 11–21. DOI: 10.15587/1729-4061.2017.95194
13. Data processing in the system of residual classes : monograph. / [V. Krasnobayev, A. Kuznetsov, A. Yanko et al.]. – ASC Academic Publishing, 2019. – 208 p. – ISBN: 978-0-9989826-6-3 (Hardback), ISBN: 978-0-9989826-7-0 (Ebook).
14. Szabo N. S. Residue Arithmetic and Its Applications to Computer Technology / N. S. Szabo, R. I. Tanaka. – New York : McGraw-Hill, 1967. – 236 p.
15. Kazymyrov O. Influence of addition modulo 2n on algebraic attacks / O. Kazymyrov, R.Oliynykov, H. Raddum // Cryptography and Communications, April 2016. – Vol. 8, Issue 2. – P. 277–289. DOI: https://doi.org/10.1007/s12095-015-0136-7
16. Krasnobayev V. A. Algorithms of data processing in the residual classes system / V. A. Krasnobayev, A. S. Yanko, S. A. Koshman // 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T). – Kharkov, 2017. – P. 117–121. DOI: 10.1109/INFOCOMMST.2017.8246363
17. Arnold M. G. The Residue Logarithmic Number System: Theory and Implementation / M. G. Arnold // 17th IEEE International Symposium on Computer Arithmetic. – 2005. – P. 196–205. DOI: 10.1109/ARITH.2005.44
18. Gavrylenko S. Y. Development of method for identification the computer system state based on the decision tree with multidimensional nodes / S. Y. Gavrylenko, V. V. Chelak, S. G. Semenov // Radio Electronics, Computer Science, Control. – 2022. – No. 2. – P. 113–121. DOI: https://doi.org/10.15588/1607-3274-2022-2-11
19. Adder based residue to binary number converters for (2/sup n/–1, 2/sup n/, 2/sup n/+1) / [Y. Wang, X. Song, M. Aboulhamid, H. Shen] // IEEE Transactions on Signal Processing. – July 2002. – Vol. 50, № 7. – P. 1772–1779. DOI: 10.1109/TSP.2002.1011216
20. A study of the residue-to-binary converters for the three-moduli sets / [Wei Wang, M. N. S. Swamy, M. O. Ahmad, Yuke Wang] // IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications. – Febuary 2003. – Vol. 50, № 2. – P. 235–243. DOI: 10.1109/TCSI.2002.808191.

DOI 10.15588/1607-3274-2022-4-2