UDC 004.222.5:681.142.01

METHOD FOR DETERMINING THE BIT GRID OVERFLOW OF A COMPUTER SYSTEM OPERATING IN THE SYSTEM OF RESIDUAL CLASSES

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.

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

Nikolsky S. B. – PhD, Associate Professor, Associate Professor of the Department of Informatics, Municipal establishment "Kharkiv humanitarian-pedagogical academy" of Kharkiv regional council, Kharkiv, Ukraine.

Kruk O. O. – Post-graduate student of the Department of Automation, Electronics and Telecommunications, National University "Yuri Kondratyuk Poltava Polytechnic".

ABSTRACT

Context. Consideration of a set of examples of practical application of the procedure for identifying overflow of the bit grid of a computer system operating in a non-positional number system in residual classes. The object of the study is the process of processing data represented in the residual class system.

Objective. The goal of the work is to consider and analyze examples of the bit grid overflow definition of a computer system when implementing the operation of adding two numbers in a system of residual classes based on the application of a method for determining the bit grid overflow, based on the use of the concept of number rank.

Method. The specificity of the functioning of a computer system in a system of residual classes requires the implementation of not only modular operations, but also requires the implementation of additional, so-called non-modular operations. Non-modular operations include the operation of determining the overflow of the bit grid of a computer system in the system of residual classes. In a non-positional number system in residual classes, implementing the process of detecting overflow of the bit grid of a computer system is a difficult task to implement. The method considered in the work for determining the overflow of the bit grid is based on the use of positional features of a non-positional code of numbers in the system of residual classes, namely the true and calculated ranks of a number. The process of determining the overflow of the result of the operation of adding two numbers in the system of residual classes has been studied, since this arithmetic operation is the main, basic operation performed by a computer system.

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

Conclusions. The main advantage of the presented method is that the process of determining the overflow of the bit grid can be carried out in the dynamics of the computing process of the computer system, i.e. without stopping the solution of the problem. This circumstance makes it possible to reduce the unproductive expenditure of the computer system in the system of residual classes. In addition, this method can be used to control the operation of adding two numbers in the residual class system. This increases the reliability of obtaining the true result of the operation of adding two numbers in the system of residual classes.

KEYWORDS: arithmetic operation of modular addition, bit grid overflow, comparison operation, computer system, non-positional code, rank of the number, system of residual classes, zeroing procedure.

ABBREVIATIONS

CS is a computer system;
MDBGO is method for determining bit grid overflow;
PFNC is a positional feature of a non-positional code;
PNS is a positional number system;
SCS is a specialized computer system;
SRC is a system in residual classes;
ZC is a zeroing constant.

NOMENCLATURE

p_i is a base (module) SRC, $(i=\overline{1,n})$;

X is a number in the SRC, represented by a set of residues x_i modulo p_i;

Y is a number in the SRC, represented by a set of residues y_i modulo p_i;

R_X is a rank of number *X*;

$R_X^{(T)}$ is a true rank of number *X*;

$R_X^{(C)}$ is a calculated rank of number *X*;

B_i is a orthogonal basis of the SRC;

e_i is a weight of the *i*-th orthogonal basis B_i;

P is a numerical range of CS in the SRC;

$R_{X+Y}^{(C)}$ is a calculated rank of the sum of two numbers *X* and *Y*;

$z^{(i)}$ is a minimum ZC for module p_i;

X_n is a zeroing number *X* (the value of number *X* as a result of the zeroing procedure);

q_i is a number of additions of type $X+z^{(i)}$;

δ_i is a known quantity that is determined sequentially in the process of transformation (in the process of zeroing) of the original number X into the number X_n, $(i=\overline{1,n})$.

DOI 10.15588/1607-3274-2024-1-21

INTRODUCTION

Solving a wide class of computational problems by the CS operating in a non-positional number system in the SRC requires additional implementation of non-modular (positional) operations. Positional operations the SRC are those operations that require knowledge of the magnitudes of numbers in the binary PNS [1].Such operations primarily include the following operations: arithmetic and algebraic comparison of numbers, determining the sign of a number, determining the location of a number on the number axis, dividing numbers, operations with the fractional part of numbers, rounding numbers, determining whether the digit grid is overflowed, diagnostics, control and correction of data in the SRC, etc [2]. Accounting for grid overflow is one of the most common problems in the field of computer science and programming. An overflow occurs when the result of a calculation cannot be represented in the current bit grid size, resulting in loss of precision and incorrect values. The consequences of overflowing the bit grid can be catastrophic. Incorrect values can lead to software crashes, data loss, unpredictable behavior, and other problems [3].

Knowledge of the fact that the bit grid is overflowed is also important when implementing in the SRC not only modular, but also various positional operations, since in the SRC the number of bits used to represent numbers is limited. For example, when adding two numbers with the same signs, its sum modulo may be greater in modulus than the maximum number that can be written with a given number of digits and the result of the addition will be incorrect. Also, the availability of information about the overflow of the bit grid is important when determining the true value of the number in the PNS [4–6].

The unsolved problem of effectively determining the overflow of the bit grid in the SRC requires the development and study of MDBGO. Therefore, the scientific task of determining the overflow of the CS bit grid in the SRC is important and relevant. The solution to this problem will contribute to the further theoretical and practical development of machine arithmetic in the non-positional number system in the SRC. This will make it possible to widely use SRC to create ultra-fast, reliable and fault-tolerant specialized CS [7].

The object of study is the process of determining the overflow of the result of the operation of adding two numbers in the SRC. The process of determining the overflow of the bit grid when implementing various operations, especially the addition operation in the CS operating in the SRC, affects various aspects of the calculations (implementation complexity, calculation accuracy).

The subject of study is the MDBGO. The method consists of a set of the following operations. The values of the calculated ranks $R_X^{(C)}$ and $R_Y^{(C)}$ of the summands X and Y are determined, and the calculated value of the rank $R_{X+Y}^{(C)}$ of the result $X+Y$ of the operation of adding two numbers is also determined. By means of orthogonal bases B_i SRC, the true value of the rank $R_{X+Y}^{(T)}$ of the result $X+Y$ of the operation of adding two numbers is determined. A comparison is made between the calculated and true ranks of numbers. A conclusion is made about the presence or absence of overflow of the result of the addition operation.

The purpose of the work is to consider and analyze examples of determining the overflow of the bit grid of the CS when implementing the operation of adding two numbers in the SRC based on the use of a method based on the use of the concept of rank of the number. To achieve the goal of the work, the following tasks are formulated and solved: to formulate the problem, to formulate a criterion for assessing the fact of overflow of the CS bit grid based on the analysis of the ranks of the summands of numbers X and Y in the SRC, to give general and specific (for a given SRC) examples of solving the problem of determining the overflow, to carry out analyze the results obtained and draw conclusions.

1 PROBLEM STATEMENT

To implement positional operations in the SRC, including determining the fact of overflow of the bit grid of the CS, various PFNC can be used [1, 8]. The rank R_X of number $X = (x_1 \| x_2 \| ... \| x_n)$ in the SRC, represented by the set of residues x_i from dividing number X itself by the set of bases $p_i\ (i = \overline{1, n})$ in the SRC, will be used as the PFNC. The SRC defines two types of rank of the number: the true $R_X^{(T)}$ and the calculated $R_X^{(C)}$ ranks of number X. The true rank $R_X^{(T)}$ is a natural number that shows how many times the numerical range $P = \prod_{i=1}^{n} p_i$ of the CS in the SRC was exceeded during the transition from the representation of the number X in the SRC to its representation in the PNS through a system of orthogonal bases of the form $B_i = \frac{e_i \cdot P}{p_i}\ (i = \overline{1, n}),$ where the value of e_i determines the weight of the i-th orthogonal basis B_i SRC [1, 9].

Let the SRC be given by its bases $p_i\ (i = \overline{1, n}).$ This SRC uniquely corresponds to a system of orthogonal bases $B_i\ (i = \overline{1, n}),$ for which the equality $X_{PNS} = \left\{ \sum_{i=1}^{n} x_i \cdot B_i \right\} \bmod P$ holds. This ratio can also be represented as $X_{PNS} = \sum_{i=1}^{n} x_i \cdot B_i - R_X^{(T)} \cdot P.$ The rank of the number, which is the result of an arithmetic operation, obtained from the ranks of numbers, is called the calculated rank of the number.

To achieve the result of the study, it is necessary to consider specific examples of identifying the fact of over-

DOI 10.15588/1607-3274-2024-1-21

flow of the CS bit grid based on the ranks of numbers X and Y in the SRC. In turn, the task of determining the overflow of the CS bit grid in the SRC is implemented by determining and comparing the calculated $R_{X+Y}^{(C)}$ and true $R_{X+Y}^{(T)}$ ranks of the number $X+Y$ in the SRC. If the condition $R_{X+Y}^{(C)} = R_{X+Y}^{(T)}$, is satisfied then it is considered that there is no overflow. Otherwise, i.e. when $R_{X+Y}^{(C)} \neq R_{X+Y}^{(T)}$, there is an overflow of the CS bit grid in the SRC.

2 REVIEW OF THE LITERATURE

One of the reasons for overflowing the bit grid is the use of large numerical values in the implemented operations. For example, when adding two numbers, if the result exceeds the maximum allowed value, an overflow occurs. Overflow can also occur when implementing other arithmetic operations. To solve the problem of eliminating the negative influence of the bit grid overflow process, there are various approaches [10–12].

At the same time, the problem of overflow of the bit grid has not yet been completely solved, since most methods for determining overflow lead to an increase in the amount of memory and slow down the calculation process in the CS [13]. Depending on the specific requirements and characteristics of the problem, the choice of method for solving the scientific problem of bit grid overflow may vary. It is important to consider both accuracy and performance and find a balance between them to achieve optimal results. Due to the relevance and unsolved nature of this problem, computer scientists are in search of effective methods and procedures for determining and eliminating the consequences of the process of overflowing the CS bit grid.

Modern publications increasingly contain various innovative methods for determining overflow when implementing various operations in a CS. Software-based architectural bound-checking based on boundary bits (bounds checking bits) that detects and prevents buffer overflows [14] results in increased memory requests to dynamically check object bindings using a boundary bit. This leads to an increase in the amount of allocated memory and affects the performance of the computer system, since one of the key aspects of increasing the performance of any software system is the efficient allocation of memory and the release of resources.

Also, to solve the problem of overflowing the bit grid, a method based on the use of modified codes is widely used [16–17]. Modification of codes consists in introducing an additional digit, which is located before the sign one. This bit is often called an overflow bit. When using various algorithms, modified codes may contain two sign bits. Article [18] presents a bidirectional overflow digital correction algorithm with a single bit redundancy used in the pipeline A/D converters. The disadvantage of all methods based on various modified codes is the expansion of the bit grid by at least one bit.

The possibility of using hardware methods to solve the overflow problem is being widely explored. In [19] considers using an N bit result integer multiplier with overflow detector indicating an N bit multiplication result and overflow status with an N bit multiplier and multiplicand input. The overflow is determined by the lower N bit result of multiplication and the number of leading sign bits of the multiplier and the multiplicand. The proposed method to prevent the bit grid from overflowing when implementing a multiplication (exponentiation) operation, with a slight decrease in performance during this operation.

To prevent overflow of the bit grid, it is necessary to take measures to optimize existing methods and develop new ones [20]. An analysis of publications in this area has shown that it is necessary to develop methods and procedures aimed at identifying and eliminating the negative consequences of overflows when implementing various operations in a computer system, which does not reduce the overall performance of the computer system, which depends on the speed of execution of these operations. This task also applies to the CS in the SRC.

3 MATERIALS AND METHODS

Since in this work the rank of a number as a PFNC was used, therefore, in the process of identifying overflow of the bit grid of the CS, it is very important to calculate the ranks of numbers X and Y. Based on this, an important task is to consider specific examples of identifying facts of overflow of the CS bit grid based on the use of the calculated ranks $R_X^{(C)}$ and $R_Y^{(C)}$ of numbers X and Y.

Let's consider the procedure for determining the calculated rank of the sum of two numbers in the SRC [1]. If two numbers $X = (x_1 \| x_2 \| ... \| x_n)$ and $Y = (y_1 \| y_2 \| ... \| y_n)$ are given in the SRC with the corresponding calculated ranks $R_X^{(C)}$ and $R_Y^{(C)}$ of numbers, then the calculated rank $R_{X+Y}^{(C)}$ of the sum of two numbers $X+Y$ is determined as follows:

$$R_{X+Y}^{(C)} = R_X^{(C)} + R_Y^{(C)} - \sum_{i=1}^{n} \left[\frac{x_i + y_i}{p_i} \right] \cdot e_i. \qquad (1)$$

Let's show the correctness of expression (1). Let's write expressions for determining the numbers X and Y in the PNS using true ranks:

$$X_{PNS} = \sum_{i=1}^{n} x_i \cdot B_i - R_X^{(T)} \cdot P, \qquad (2)$$

$$Y_{PNS} = \sum_{i=1}^{n} y_i \cdot B_i - R_Y^{(T)} \cdot P. \qquad (3)$$

Let's add two expressions (2) and (3) and get:

DOI 10.15588/1607-3274-2024-1-21

$$X_{PNS}+Y_{PNS}=\sum_{i=1}^{n}x_i\cdot B_i-R_X^{(T)}\cdot P+\sum_{i=1}^{n}y_i\cdot B_i-R_Y^{(T)}\cdot P,$$

or

$$X_{PNS}+Y_{PNS}=\sum_{i=1}^{n}(x_i+y_i)\cdot B_i-(R_X^{(T)}+R_Y^{(T)})\cdot P. \qquad (4)$$

On the other hand, based on the rule for calculating the sum of two numbers in the SRC for each corresponding SRC base can be written that:

$$X+Y=\left\{\left(x_1+y_1-\left[\frac{x_1+y_1}{p_1}\right]\cdot e_1\right),\right.$$
$$\left(x_2+y_2-\left[\frac{x_2+y_2}{p_2}\right]\cdot e_2\right),\dots \qquad (5)$$
$$\left.\dots,\left(x_n+y_n-\left[\frac{x_n+y_n}{p_n}\right]\cdot e_n\right)\right\}$$

Expression (5) when using expression (2) can be represented as:

$$X+Y=\sum_{i=1}^{n}\left(\left\{\left(x_i+y_i\right)-\left[\frac{x_i+y_i}{p_i}\right]\cdot e_i\right\}\right)\cdot B_i-R_{X+Y}^{(T)}\cdot P. \qquad (6)$$

Let's transform expression (6), taking into account the fact that the orthogonal basis SRC is represented as $B_i=\frac{e_i\cdot P}{p_i}$. As a result, the following expression can be obtained:

$$X+Y=\sum_{i=1}^{n}\left(x_i+y_i\right)\cdot B_i-\sum_{i=1}^{n}\left[\frac{x_i+y_i}{p_i}\right]\cdot e_i\cdot B_i-R_{X+Y}^{(T)}\cdot P, \qquad (7)$$

or

$$X+Y=\sum_{i=1}^{n}\left(x_i+y_i\right)\cdot B_i-\sum_{i=1}^{n}\left[\frac{x_i+y_i}{p_i}\right]\cdot e_i\cdot\frac{e_i\cdot P}{p_i}-R_{X+Y}^{(T)}\cdot P. \qquad (8)$$

Let's compare the right-hand sides of expressions (4) and (8) to check the correctness of expression (1):

$$\sum_{i=1}^{n}\left(x_i+y_i\right)\cdot B_i-\left(R_X^{(T)}+R_Y^{(T)}\right)\cdot P=$$
$$=\sum_{i=1}^{n}\left(x_i+y_i\right)\cdot B_i-\sum_{i=1}^{n}\left[\frac{x_i+y_i}{p_i}\right]\cdot e_i\cdot P-R_{X+Y}^{(T)}\cdot P,$$

that is, we have that expression (1) is satisfied (fairly).

Expression (1) is the main analytical expression that allows us to determine the calculated rank of the sum of two numbers *X* and *Y* from the values of the calculated ranks of the summands *X* and *Y*.

It's obvious that:

if $x_i+y_i\ge p_i$ then the integer part of the expression is equal to $\left[\frac{x_i+y_i}{p_i}\right]=1;$

if $x_i+y_i<p_i$ then the integer part of the expression is equal to $\left[\frac{x_i+y_i}{p_i}\right]=0.$

The procedure for determining the calculated rank $R_X^{(C)}$ of number $X=(x_1\|x_2\|...\|x_n)$ is as follows. First, it needs to present the original number $X=(x_1\|x_2\|...\|x_n)$ to a zeroable number of the form $X_n=(0\|0\|...\|0)$. To do this, to the initial number $X=(x_1\|x_2\|...\|x_n)$ in the SRC, the rank $R_X^{(C)}$ of which must be determined, the so-called ZC, in the form of minimum numbers of the form $z^{(i)}=(0\|0\|...\|0\|z_i\|z_{i+1}\|...\|z_n)$ $(i=\overline{1,n})$, are sequentially added. In this case, this value in the PNS is equal to the value $z_{PNS}^{(i)}=p_1\cdot p_2\cdot...\cdot p_{i-1}$.

In particular, we find that the ZC has the following form:

$$z^{(1)}=\min(z_1^{(1)}\|z_2^{(1)}\|...\|z_n^{(1)})=(1\|1\|...\|1),$$
$$z^{(2)}=\min(0\|z_2^{(2)}\|...\|z_n^{(2)})=(0\|p_1\|p_2\|...\|p_{i-1}),$$
$$z^{(3)}=\min(0\|0\|z_3^{(3)}\|z_4^{(3)}\|...\|z_n^{(3)})=\{(0\|0\|p_1\cdot p_2(\bmod p_3)\|$$
$$\|p_1\cdot p_2(\bmod p_4)\|...\|p_1\cdot p_2(\bmod p_n)\},$$

etc., where $z^{(n)}=(0\|0\|...\|0\|z_n)$.

Let's show the procedure for obtaining the value of $X_n=(0\|0\|...\|0)$. At the beginning of the procedure, let's add the ZC $z^{(1)}=(z_1^{(1)}\|z_2^{(1)}\|...\|z_n^{(1)})$ to the initial number X as many times as necessary to satisfy the condition $x_1=0$. Let this require q_1 additions of type $X+z^{(1)}$. In this case, we get that $X_1=X+q_1\cdot z^{(1)}$. As a result, the resulting number X_1 has an intermediate calculated rank $R_{X_1}^{(C)}$ (the intermediate calculated rank is the calculated rank of the number, which is sequentially formed in the process of obtaining the value $X_n=(0\|0\|...\|0)$).

Then, we obtain that $R_{X_1}^{(C)}=R_X^{(C)}+\delta_1$, where δ_1 is a known value. Next, we add q_2 times the value of the ZC $z^{(2)}=(0\|z_2^{(2)}\|...\|z_n^{(2)})$ with the number X_1 until we obtain a zero residue to the base p_2, i.e. we obtain $x_2=0$. So we have a number $X_2=X_1+q_2\cdot z^{(2)}$ with an intermediate rank $R_{X_2}^{(C)}=R_{X_1}^{(C)}+\delta_2$, where δ_2 is a known value. The algorithm for obtaining the number $X_n=(0\|0\|...\|0)$ can be represented by the following expressions:

DOI 10.15588/1607-3274-2024-1-21

$$\begin{cases} X_1 = X + q_1 \cdot z^{(1)}, R_{X_1}^{(C)} = R_X^{(C)} + \delta_1; \\ X_2 = X_1 + q_2 \cdot z^{(2)}, R_{X_2}^{(C)} = R_{X_1}^{(C)} + \delta_2; \\ \ldots \\ X_i = X_{i-1} + q_i \cdot z^{(i)}, R_{X_i}^{(C)} = R_{X_{i-1}}^{(C)} + \delta_i; \\ \ldots \\ X_n = X_{n-1} + q_n \cdot z^{(n)}, R_{X_n}^{(C)} = R_{X_{n-1}}^{(C)} + \delta_n. \end{cases} \quad (9)$$

Continuing the procedure for all remainders of the number X, the result is the number $X_n = (0 \| 0 \| \ldots \| 0) = P$. In accordance with expression (2) we have that:

$$X_{PNS} = \sum_{i=1}^{n} x_i \cdot B_i - R_X^{(T)} \cdot P,$$

$$X_n = \sum_{i=1}^{n} x_i \cdot B_i - R_{X_n}^{(T)} \cdot P,$$

$$(0 \| 0 \| \ldots \| 0) = \sum_{i=1}^{n} x_i \cdot B_i - R_{X_n}^{(T)} \cdot P,$$

$$P = 0 - R_{X_n}^{(T)} \cdot P,$$

$$R_{X_n}^{(T)} = -1. \quad (10)$$

Thus, the true rank $R_{X_n}^{(T)}$ of the zeroable number $X_n = (0 \| 0 \| \ldots \| 0)$ is equal to – 1. On the other hand, it was shown in expression (9) that the calculated rank $R_{X_n}^{(C)}$ of the zeroable number X_n is equal to $R_{X_{n-1}}^{(C)} + \delta_n$. Since the value of the calculated rank $R_{X_n}^{(C)}$ must coincide with the true rank $R_{X_n}^{(T)}$, then the last expression of the ratio (9) and (10) must coincide:

$$R_{X_n}^{(C)} = R_{X_n}^{(T)} \Rightarrow R_{X_n}^{(C)} = -1 \quad (11)$$

or expression (11) can be written as:

$$R_{X_{n-1}}^{(C)} + \delta_n = -1 \Rightarrow R_{X_{n-1}}^{(C)} = -1 - \delta_n. \quad (12)$$

4 EXPERIMENTS

The theoretical basis for creating an experimental research base is scientific material, which is presented in the relevant sections of number theory and is also presented as the result of the proof of the Chinese remainder theorem [21]. In this case, the initial data for conducting the experiment are presented in the form of a set of bases (modules) of the SRC. SRC bases are a set of mutually prime numbers. As an experiment, this work presents the content and discusses the description of the structure of the method for determining the overflow of the CS bit grid in the SRC.

The general scheme of the experiment to determine the overflow of the bit grid of the CS, when implementing the operation of adding two numbers in the SRC [22], is presented in the following form:

1. Using the zeroing procedure based on the minimum ZCs $z^{(i)}$ SRC, the values of the calculated ranks $R_X^{(C)}$ and $R_Y^{(C)}$ of numbers are determined. A comparison is made between the calculated and true ranks of numbers.
2. Using the minimum ZCs $z^{(i)}$ SRC and calculated ranks values $R_X^{(C)}$ and $R_Y^{(C)}$ of numbers X and Y, according to expression (1), the calculated value $R_{X+Y}^{(C)}$ of the rank of the result X+Y of the operation of adding two numbers is determined.
3. By means of orthogonal bases B_i SRC, the true value of the rank $R_{X+Y}^{(T)}$ of the result of the operation of adding two numbers X+Y is determined.
4. A comparison is made between the calculated and true ranks of numbers. A conclusion is made about the fact of the bit grid overflow of the result of the addition operation according to the rank comparison criterion.

The work, as an experiment, provides a set of specific examples of the use of MDBGO. The results of the experiment showed the practical significance of the method under consideration. In addition, based on the use of the main results of the experiments, the State Patent of Ukraine for a utility model No. 129125, G06F 11/08 "Device for monitoring the result A+B of the addition of two numbers A and B in the system of residual classes" (published 10.25.2018, Bull. No. 20) was obtained, authors: Krasnobayev V. A., Yanko A. S. et al. This device contains blocks for determining the calculated and true values of the ranks of the result of the operation of adding two numbers. Also this device also contains a block for comparing the calculated and true ranks of numbers in the SRC. The invention is based on the use of MDBGO. The purpose of this invention is to reduce the control time of the modular operation of addition of two numbers presented in the SRC. The goal is achieved by combining in time data processing operations in blocks for determining the calculated and true values of the ranks of numbers. This increases the efficiency of monitoring the implementation of the modular operation of adding of numbers. Thus, the above described allows, in addition to implementing the main function, to additionally use MDBGO to control the procedure for adding numbers in the SRC. This indicates the versatility of using the method discussed in the article. The presence of a patent confirms the global novelty and practical significance of some of the scientific results obtained in the article. Some results obtained in the article are an undoubted contribution to the theory and practice of non-positional machine arithmetic. The results obtained in the article can be used when creating a CS in the SRC.

5 RESULTS

In accordance with the procedure for determining the rank of a number, let's consider examples of determining the calculated rank $R_X^{(C)}$ of number $X = (x_1 \| x_2 \| \ldots \| x_n)$ presented in a specific SRC. To the initial number $X = (x_1 \| x_2 \| \ldots \| x_n)$ in the SRC, the calculated rank $R_X^{(C)}$

DOI 10.15588/1607-3274-2024-1-21

of which must be determined, let's successively add the minimum ZCs $z^{(i)}$ until we ultimately obtain the zeroable number $X_n = (0 \| 0 \| ... \| 0)$, the intermediate calculated rank $R^{(C)}_{X_n}$, which from expression (11) is equal to $R^{(C)}_{X_n} = -1$. Next, using orthogonal bases, the true rank of the number is determined.

Let's give examples of determining the calculated rank $R^{(C)}_X$ of a number *X*. Table 1 presents the SRC bases $\{p_i\}$, $i = \overline{1,3}$, orthogonal bases B_i of the bases and its weights e_i. In Table 2, for a given SRC, the minimum ZCs $z^{(i)}$ and its ranks $R_{z^{(i)}}$ are given. For the SRC under consideration, the volume of the range of representable numbers is equal to $P = \prod_{i=1}^{3} p_i = 3 \cdot 5 \cdot 7 = 105$.

Table 1 – Values of the SRC bases and orthogonal bases

$p_1 = 3$	$p_2 = 5$	$p_3 = 7$
$e_1 = 2$	$e_2 = 1$	$e_3 = 1$
$B_1 = 70$	$B_2 = 21$	$B_3 = 15$

Table 2 – Values of minimum ZCs and its ranks

$z^{(1)} = (1 \| 1 \| 1)$	$z^{(2)} = (0 \| 3 \| 3)$	$z^{(3)} = (0 \| 0 \| 1)$
$R_{z^{(1)}} = 1$	$R_{z^{(2)}} = 1$	$R_{z^{(3)}} = 0$

The ranks $R_{z^{(i)}}$ of the minimum ZCs $z^{(i)}$ are calculated in advance using expression (2). Let's determine the values of the minimum constants for the SRC specified in Table 1:

$$z^{(1)} = (1 \| 1 \| 1) = 1 \cdot B_1 + 1 \cdot B_2 + 1 \cdot B_3 = 1 \cdot 70 + 1 \cdot 21 + 1 \cdot 15 =$$
$$= (70 + 21 + 15) \bmod 105 = 106 \bmod 105 =$$
$$= \sum_{i=1}^{n} z_i \cdot B_i - R_{z^{(1)}} \cdot P = 106 - R_{z^{(1)}} \cdot P = 106 - 1 \cdot 105.$$

to comply with the above equality, it comes out to $R_{z^{(1)}} = 1$ (Table 2).

$$z^{(2)} = (0 \| 3 \| 3) = 0 \cdot B_1 + 3 \cdot B_2 + 3 \cdot B_3 =$$
$$= 0 \cdot 70 + 3 \cdot 21 + 3 \cdot 15 = (0 + 63 + 45) \bmod 105 = 108 \bmod 105 =$$
$$= \sum_{i=1}^{n} z_i \cdot B_i - R_{z^{(2)}} \cdot P = 108 - R_{z^{(2)}} \cdot P = 108 - 1 \cdot 105.$$

to comply with the above equality, it comes out to $R_{z^{(2)}} = 1$ (Table 2).

$$z^{(3)} = (0 \| 0 \| 1) = 0 \cdot B_1 + 0 \cdot B_2 + 1 \cdot B_3 =$$
$$= 0 \cdot 70 + 0 \cdot 21 + 1 \cdot 15 = (0 + 0 + 15) \bmod 105 = 15 \bmod 105 =$$
$$= \sum_{i=1}^{n} z_i \cdot B_i - R_{z^{(3)}} \cdot P = 15 - R_{z^{(3)}} \cdot 105 = 15 - 0 \cdot 105,$$

to comply with the above equality, it comes out to $R_{z^{(3)}} = 0$ (Table 2).

Example 1. Determine the calculated rank $R^{(C)}_X$ of the number $X = (2 \| 1 \| 1) = 71$.

First stage. Determination of the calculated rank of the number in the SRC.

Let's reset the residue $x_1 = 2$ to zero according to the first module $p_1 = 3$. Let's add the number *X* and minimum ZC $z^{(1)} = (1 \| 1 \| 1)$ and get:

$$X_1 = X + z^{(1)} = (2 \| 1 \| 1) + (1 \| 1 \| 1) = (0 \| 2 \| 2).$$

The rank of the sum will be determined by expression (1), where instead of the rank of the number *Y* the value of the rank of the minimum ZC $z^{(i)}$ will be used:

$$R^{(C)}_{X_i} = R^{(C)}_{X_{i-1}} + R_{z^{(i)}} - \sum_{i=1}^{3} \left[\frac{x_i + z_j^{(i)}}{p_i} \right] \cdot e_i, \qquad (13)$$

where $[k]$ is the integer part of the number *k*, not less than it; $R_{z^{(i)}}$ is the calculated rank of minimum ZC $z^{(i)}$ (Table 2); $z_j^{(i)}$ is the value of the *j*-th residue, $j = \overline{1,n}$ (in this case $j = \overline{1,3}$) of the *i*-th minimum ZC.

Based on expression (13), the following calculated value of the rank of the number X_1 can be obtained:

$$R^{(C)}_{X_1} = R^{(C)}_X + R_{z^{(1)}} - \sum_{i=1}^{3} \left[\frac{x_i + z_j^{(1)}}{p_i} \right] \cdot e_i = R^{(C)}_X + R_{z^{(1)}} -$$
$$- \left\{ \left[\frac{x_1 + z_1^{(1)}}{p_1} \right] \cdot e_1 + \left[\frac{x_2 + z_2^{(1)}}{p_2} \right] \cdot e_2 + \left[\frac{x_3 + z_3^{(1)}}{p_3} \right] \cdot e_3 \right\} =$$
$$= \left(R^{(C)}_X + 1 \right) - \left\{ \left[\frac{2+1}{3} \right] \cdot 2 + \left[\frac{1+1}{5} \right] \cdot 1 + \left[\frac{1+1}{7} \right] \cdot 1 \right\} =$$
$$= R^{(C)}_X + 1 - (1 \cdot 2 + 0 \cdot 1 + 0 \cdot 1) = R^{(C)}_X + 1 - 2 = R^{(C)}_X - 1.$$

In this case, there was one transition through the first base p_1 (expression (9)).

Let's reset the residue $x_2 = 2$ to zero according to the second module $p_2 = 5$ of the number $X_1 = (0 \| 2 \| 2)$. Let's add the number X_1 and minimum ZC $z^{(2)} = (0 \| 3 \| 3)$ and get:

$$X_2 = X_1 + z^{(2)} = (0 \| 2 \| 2) + (0 \| 3 \| 3) = (0 \| 0 \| 5).$$

The calculated rank of the number X_2 is determined as:

DOI 10.15588/1607-3274-2024-1-21

$$R_{X_2}^{(C)} = R_{X_1}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(2)}}{p_i}\right]\cdot e_i \ = \left(R_X^{(C)} - 1\right) +$$
$$+R_{z^{(2)}} - \left\{\left[\frac{x_1 + z_1^{(2)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2 + z_2^{(2)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3 + z_3^{(2)}}{p_3}\right]\cdot e_3\right\} =$$
$$= \left(R_X^{(C)} - 1\right) + 1 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{2+3}{5}\right]\cdot 1 + \left[\frac{2+3}{7}\right]\cdot 1\right\} =$$
$$= R_X^{(C)} - 1 + 1 - 0\cdot 2 - 1\cdot 1 - 0\cdot 1 = R_X^{(C)} - 1.$$

In this case, there was one transition through the second base p_2.

Let's reset the residue $x_3 = 5$ to zero according to the third module $p_3 = 7$ of the number $X_2 = (0 \,||\, 0 \,||\, 5)$. Let's add the number X_2 and minimum ZC $z^{(3)} = (0 \,||\, 0 \,||\, 1)$ and get:

$$X_3 = X_2 + z^{(3)} = (0 \,||\, 0 \,||\, 5) + (0 \,||\, 0 \,||\, 1) = (0 \,||\, 0 \,||\, 6).$$

The calculated rank of the number X_3 is determined as:

$$R_{X_3}^{(C)} = R_{X_2}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(3)}}{p_i}\right]\cdot e_i \ = \left(R_X^{(C)} - 1\right) +$$
$$+R_{z^{(3)}} - \left\{\left[\frac{x_1 + z_1^{(3)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2 + z_2^{(3)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3 + z_3^{(3)}}{p_3}\right]\cdot e_3\right\} =$$
$$= \left(R_X^{(C)} - 1\right) + 0 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{0+0}{5}\right]\cdot 1 + \left[\frac{5+1}{7}\right]\cdot 1\right\} =$$
$$= R_X^{(C)} - 1 + 0 - 0 - 0 - 0 = R_X^{(C)} - 1.$$

Since the residue $x_3 = 6$ of the number $X_3 = (0 \,||\, 0 \,||\, 6)$ has not been reset to zero, let's add the value $z^{(3)}$ again:

$$X_n = X_3 + z^{(3)} = (0 \,||\, 0 \,||\, 6) + (0 \,||\, 0 \,||\, 1) = (0 \,||\, 0 \,||\, 0).$$

The calculated rank of the zeroable number X_n is determined as:

$$R_{X_n}^{(C)} = R_{X_3}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(3)}}{p_i}\right]\cdot e_i \ = \left(R_X^{(C)} - 1\right) +$$
$$+R_{z^{(3)}} - \left\{\left[\frac{x_1 + z_1^{(3)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2 + z_2^{(3)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3 + z_3^{(3)}}{p_3}\right]\cdot e_3\right\} =$$
$$= \left(R_X^{(C)} - 1\right) + 0 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{0+0}{5}\right]\cdot 1 + \left[\frac{6+1}{7}\right]\cdot 1\right\} =$$
$$= R_X^{(C)} - 1 + 0 - 0 - 0 - 1\cdot 1 = R_X^{(C)} - 2.$$

In this case, there was one transition through the third base p_3.

In accordance with expression (11) we have that:

$$R_{X_n}^{(C)} = -1 \rightarrow R_X^{(C)} - 2 = -1 \rightarrow R_X^{(C)} = 1.$$

Second stage. Determination of the true rank of the number in the SRC.

The true rank $R_X^{(T)}$ of number X is determined using orthogonal bases B_i (Table 1) and expression (2):

$$X = (2 \,||\, 1 \,||\, 1) = x_1 \cdot B_1 + x_2 \cdot B_2 + x_3 \cdot B_3 =$$
$$= 2\cdot B_1 + 1\cdot B_2 + 1\cdot B_3 = 2\cdot 70 + 1\cdot 21 + 1\cdot 15 =$$
$$= (2\cdot 70 + 1\cdot 21 + 1\cdot 15) \bmod 105 = 176 \bmod 105 =$$
$$= X_{PNS} = \sum_{i=1}^{n} x_i \cdot B_i - R_X^{(T)} \cdot P =$$
$$= 176 \bmod 105 = 176 - R_X^{(T)} \cdot P =$$
$$= 176 - 1\cdot 105 = 176 - 105 = 71.$$

Thus, the true of the number $X = (2 \,||\, 1 \,||\, 1) = 71$ is $R_X^{(T)} = 1$.

Third stage. Checking the reliability of obtaining the rank of the number in the SRC.

Let's compare the calculated $R_X^{(C)}$ and true $R_X^{(T)}$ ranks of the number X. Obviously, $R_X^{(C)} = R_X^{(T)} = 1$.

Conclusions. So, since the calculated $R_X^{(C)}$ and true $R_X^{(T)}$ ranks of the number X are equal, then the calculated rank $R_X^{(C)}$ is determined correctly.

Example 2. Determine the calculated rank $R_X^{(C)}$ of the number $X = (1 \,||\, 1 \,||\, 5) = 61$.

First stage. Determination of the calculated rank of the number in the SRC.

Let's reset the residue $x_1 = 1$ to zero according to the first module $p_1 = 3$. Let's add the number X and minimum ZC $z^{(1)} = (1 \,||\, 1 \,||\, 1)$ and get:

$$X_1 = X + z^{(1)} = (1 \,||\, 1 \,||\, 5) + (1 \,||\, 1 \,||\, 1) = (2 \,||\, 2 \,||\, 6).$$

The calculated rank of the number X_1 will be determined by expression (13):

$$R_{X_1}^{(C)} = R_X^{(C)} + R_{z^{(1)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(1)}}{p_i}\right]\cdot e_i = R_X^{(C)} + R_{z^{(1)}} -$$
$$- \left\{\left[\frac{x_1 + z_1^{(1)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2 + z_2^{(1)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3 + z_3^{(1)}}{p_3}\right]\cdot e_3\right\} =$$
$$= R_X^{(C)} + 1 - \left\{\left[\frac{1+1}{3}\right]\cdot 2 + \left[\frac{1+1}{5}\right]\cdot 1 + \left[\frac{5+1}{7}\right]\cdot 1\right\} =$$
$$= R_X^{(C)} + 1 - 0\cdot 2 - 0\cdot 1 - 0\cdot 1 = R_X^{(C)} + 1.$$

Since the residue x_1 of the number has not been reset to zero, let's add the value $z^{(1)}$ again:

$$X_2 = X_1 + z^{(1)} = (2 \,||\, 2 \,||\, 6) + (1 \,||\, 1 \,||\, 1) = (0 \,||\, 3 \,||\, 0).$$

DOI 10.15588/1607-3274-2024-1-21

The calculated rank of the number X_2 is determined as:

$$R_{X_2}^{(C)} = R_{X_1}^{(C)} + R_{z^{(1)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(1)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)}+1\right) + R_{z^{(1)}} -$$

$$-\left\{\left[\frac{x_1+z_1^{(1)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2+z_2^{(1)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3+z_3^{(1)}}{p_3}\right]\cdot e_3\right\} =$$

$$= \left(R_X^{(C)}+1\right) + 1 - \left\{\left[\frac{2+1}{3}\right]\cdot 2 + \left[\frac{2+1}{5}\right]\cdot 1 + \left[\frac{6+1}{7}\right]\cdot 1\right\} =$$

$$= R_X^{(C)} + 1 + 1 - 1\cdot 2 - 0\cdot 1 - 1\cdot 1 = R_X^{(C)} - 1.$$

In this case, two transitions took place through the first base p_1 and through the third base p_3.

Let's reset the residue $x_2 = 2$ to zero according to the second module $p_2 = 5$ of the number $X_2 = (0\,||\,3\,||\,0)$. Let's add the number X_2 and minimum ZC $z^{(2)} = (0\,||\,3\,||\,3)$ and get:

$$X_3 = X_2 + z^{(2)} = (0\,||\,3\,||\,0) + (0\,||\,3\,||\,3) = (0\,||\,1\,||\,3).$$

The calculated rank of the number X_3 is determined as:

$$R_{X_3}^{(C)} = R_{X_2}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(2)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)}-1\right) +$$

$$+R_{z^{(2)}} - \left\{\left[\frac{x_1+z_1^{(2)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2+z_2^{(2)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3+z_3^{(2)}}{p_3}\right]\cdot e_3\right\} =$$

$$= \left(R_X^{(C)}-1\right) + 1 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{3+3}{5}\right]\cdot 1 + \left[\frac{0+3}{7}\right]\cdot 1\right\} =$$

$$= R_X^{(C)} - 1 + 1 - 0\cdot 2 - 1\cdot 1 - 0\cdot 1 = R_X^{(C)} - 1$$

So, since the residue x_2 of the number $X_3 = (0\,||\,1\,||\,3)$ has not been reset to zero, then let's add the value $z^{(2)}$ again:

$$X_4 = X_3 + z^{(2)} = (0\,||\,1\,||\,3) + (0\,||\,3\,||\,3) = (0\,||\,4\,||\,6).$$

The calculated rank of the number X_4 is determined as:

$$R_{X_4}^{(C)} = R_{X_3}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(2)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)}-1\right) +$$

$$+R_{z^{(2)}} - \left\{\left[\frac{x_1+z_1^{(2)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2+z_2^{(2)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3+z_3^{(2)}}{p_3}\right]\cdot e_3\right\} =$$

$$= \left(R_X^{(C)}-1\right) + 1 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{1+3}{5}\right]\cdot 1 + \left[\frac{3+3}{7}\right]\cdot 1\right\} =$$

$$= R_X^{(C)} - 1 + 1 - 0\cdot 2 - 0\cdot 1 - 0\cdot 1 = R_X^{(C)}.$$

There are no transitions along the bases. Since the residue x_2 of the number $X_4 = (0\,||\,4\,||\,6)$ has not been reset to zero, let's add the value $z^{(2)}$ again:

$$X_5 = X_4 + z^{(2)} = (0\,||\,4\,||\,6) + (0\,||\,3\,||\,3) = (0\,||\,2\,||\,2).$$

The calculated rank of the number X_5 is determined as:

$$R_{X_5}^{(C)} = R_{X_4}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(2)}}{p_i}\right]\cdot e_i =$$

$$= R_X^{(C)} + 1 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{4+3}{5}\right]\cdot 1 + \left[\frac{6+3}{7}\right]\cdot 1\right\} =$$

$$= R_X^{(C)} + 1 - 0\cdot 2 - 1\cdot 1 - 1\cdot 1 = R_X^{(C)} + 1 - 2 = R_X^{(C)} - 1.$$

Since the residue x_2 has not been reset to zero, the operation of adding two numbers X_5 and $z^{(2)}$ is implemented again:

$$X_6 = X_5 + z^{(2)} = (0\,||\,2\,||\,2) + (0\,||\,3\,||\,3) = (0\,||\,0\,||\,5).$$

The calculated rank of the number X_6 is determined as:

$$R_{X_6}^{(C)} = R_{X_5}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(2)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)}-1\right) +$$

$$+1 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{2+3}{5}\right]\cdot 1 + \left[\frac{2+3}{7}\right]\cdot 1\right\} =$$

$$= R_X^{(C)} - 1 + 1 - 0\cdot 2 - 1\cdot 1 - 0\cdot 1 = R_X^{(C)} - 1.$$

In this case, there was one transition through the second base p_2.

Let's reset the residue $x_3 = 5$ to zero according to the third module $p_3 = 7$ of the number $X_6 = (0\,||\,0\,||\,5)$. Let's add the number X_6 and minimum ZC $z^{(3)} = (0\,||\,0\,||\,1)$ and get:

$$X_7 = X_6 + z^{(3)} = (0\,||\,0\,||\,5) + (0\,||\,0\,||\,1) = (0\,||\,0\,||\,6).$$

The calculated rank of the number X_7 is determined as:

$$R_{X_7}^{(C)} = R_{X_6}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(3)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)}-1\right) +$$

$$+R_{z^{(3)}} - \left\{\left[\frac{x_1+z_1^{(3)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2+z_2^{(3)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3+z_3^{(3)}}{p_3}\right]\cdot e_3\right\} =$$

$$= \left(R_X^{(C)}-1\right) + 0 - \left(\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{0+0}{5}\right]\cdot 1 + \left[\frac{5+1}{7}\right]\cdot 1\right) =$$

$$= R_X^{(C)} - 1 + 0 - 0\cdot 2 - 0\cdot 1 - 0\cdot 1 = R_X^{(C)} - 1.$$

DOI 10.15588/1607-3274-2024-1-21

Since the residue x_3 has not been reset to zero, the operation of adding two numbers is implemented again:

$$X_n = X_7 + z^{(3)} = (0 \| 0 \| 6) + (0 \| 0 \| 1) = (0 \| 0 \| 0).$$

The calculated rank of the zeroable number X_n is determined as:

$$R_{X_n}^{(C)} = R_{X_7}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3} \left[\frac{x_i + z_j^{(3)}}{p_i} \right] \cdot e_i =$$
$$= \left(R_X^{(C)} - 1 \right) + 0 - \left(\left[\frac{0+0}{3} \right] \cdot 2 + \left[\frac{0+0}{5} \right] \cdot 1 + \left[\frac{6+1}{7} \right] \cdot 1 \right) =$$
$$= R_X^{(C)} - 1 + 0 - 0 \cdot 2 - 0 \cdot 1 - 1 \cdot 1 = R_X^{(C)} - 2.$$

In this case, there was one transition through the third base p_3.

In accordance with expression (11) we have that:

$$R_{X_n}^{(C)} = -1 \to R_X^{(C)} - 2 = -1 \to R_X^{(C)} = 1.$$

Second stage. Determination of the true rank of the number in the SRC.

The true rank $R_X^{(T)}$ of number X is determined using orthogonal bases B_i (Table 1) and expression (2):

$$X = (1 \| 1 \| 5) = x_1 \cdot B_1 + x_2 \cdot B_2 + x_3 \cdot B_3 =$$
$$= 1 \cdot B_1 + 1 \cdot B_2 + 5 \cdot B_3 = 1 \cdot 70 + 1 \cdot 21 + 5 \cdot 15 =$$
$$= (1 \cdot 70 + 1 \cdot 21 + 5 \cdot 15) \bmod 105 =$$
$$= 166 \bmod 105 = 166 - R_X^{(T)} \cdot P =$$
$$= 166 - 1 \cdot 105 = 61.$$

Thus, the true of the number $X = (1 \| 1 \| 5) = 71$ is $R_X^{(T)} = 1$.

Third stage. Checking the reliability of obtaining the rank of the number in the SRC.

Let's compare the calculated $R_X^{(C)}$ and true $R_X^{(T)}$ ranks of the number X. Obviously, $R_X^{(C)} = R_X^{(T)} = 1$.

Conclusions. So, since the calculated $R_X^{(C)}$ and true $R_X^{(T)}$ ranks of the number X are equal, then the calculated rank $R_X^{(C)}$ is determined correctly.

Example 3. Carry out control of the arithmetic operation of addition of two numbers $X = (x_1 \| x_2 \| x_3) = (2 \| 4 \| 4) = 74$ and $Y = (y_1 \| y_2 \| y_3) = (2 \| 3 \| 1) = 8$ presented in the SRC.

In accordance with the procedure described above for determining the rank of a number in the SRC, it is initially necessary to determine the calculated ranks $R_X^{(C)}$ and $R_Y^{(C)}$ of the summands of the numbers X and Y.

First, in accordance with the control method, let's determine the calculated rank $R_X^{(C)}$ of the number $X = (2 \| 4 \| 4) = 74$.

Let's zero the number X to the first base by adding the value of minimum ZC $z^{(1)}$:

$$X_1 = X + z^{(1)} = (2 \| 4 \| 4) + (1 \| 1 \| 1) = (0 \| 5 \| 5).$$

Using expression (13), we obtain:

$$R_{X_1}^{(C)} = R_X^{(C)} + R_{z^{(1)}} - \sum_{i=1}^{3} \left[\frac{x_i + z_j^{(1)}}{p_i} \right] \cdot e_i = R_X^{(C)} + R_{z^{(1)}} -$$
$$- \left\{ \left[\frac{x_1 + z_1^{(1)}}{p_1} \right] \cdot e_1 + \left[\frac{x_2 + z_2^{(1)}}{p_2} \right] \cdot e_2 + \left[\frac{x_3 + z_3^{(1)}}{p_3} \right] \cdot e_3 \right\} =$$
$$= R_X^{(C)} + 1 - \left\{ \left[\frac{2+1}{3} \right] \cdot 2 + \left[\frac{4+1}{5} \right] \cdot 1 + \left[\frac{4+1}{7} \right] \cdot 1 \right\} =$$
$$= R_X^{(C)} + 1 - 1 \cdot 2 - 1 \cdot 1 - 0 \cdot 1 = R_X^{(C)} + 1 - 3 = R_X^{(C)} - 2.$$

So, since the residue $x_3 = 5$ has not been reset to zero, then let's add the value $z^{(3)} = (0 \| 0 \| 1)$:

$$X_2 = X_1 + z^{(3)} = (0 \| 5 \| 5) + (0 \| 0 \| 1) = (0 \| 0 \| 6).$$

Using expression (13) let's determine the calculated rank of the number $X_2 = (0 \| 0 \| 6)$:

$$R_{X_2}^{(C)} = R_{X_1}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3} \left[\frac{x_i + z_j^{(3)}}{p_i} \right] \cdot e_i =$$
$$= \left(R_X^{(C)} - 2 \right) + R_{z^{(3)}} - \left\{ \left[\frac{0+0}{3} \right] \cdot 2 + \left[\frac{0+0}{5} \right] \cdot 1 + \left[\frac{5+1}{7} \right] \cdot 1 \right\} =$$
$$= R_X^{(C)} - 2 + 0 - 0 \cdot 2 - 0 \cdot 1 - 0 \cdot 1 = R_X^{(C)} - 2.$$

So, since the residue $x_3 = 6$ has not been reset to zero, then let's add the value $z^{(3)}$ again:

$$X_n = X_2 + z^{(3)} = (0 \| 0 \| 6) + (0 \| 0 \| 1) = (0 \| 0 \| 0).$$

For the zeroable number $X_n = (0 \| 0 \| 0)$ according to expression (13) we have the calculated rank equal to:

$$R_{X_n}^{(C)} = R_{X_2}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3} \left[\frac{x_i + z_j^{(3)}}{p_i} \right] \cdot e_i =$$
$$= \left(R_X^{(C)} - 2 \right) + R_{z^{(3)}} - \left\{ \left[\frac{0+0}{3} \right] \cdot 2 + \left[\frac{0+0}{5} \right] \cdot 1 + \left[\frac{6+1}{7} \right] \cdot 1 \right\} =$$
$$= R_X^{(C)} - 2 + 0 - 0 \cdot 2 - 0 \cdot 1 - 1 \cdot 1 = R_X^{(C)} - 3.$$

Based on expression (11), we can determine the calculated rank $R_X^{(C)}$ of the number $X = (2 \| 4 \| 4) = 74$:

DOI 10.15588/1607-3274-2024-1-21

$$R_{X_n}^{(C)} = -1 \rightarrow R_X^{(C)} - 3 = -1 \rightarrow R_X^{(C)} = 2.$$

Checking. Let's calculate the true rank $R_X^{(T)}$ of number $X = (2 \| 4 \| 4) = 74$ using orthogonal bases B_i SRC (Table 1). We have that in the PNS:

$$X_{PNS} = x_1 \cdot B_1 + x_2 \cdot B_2 + x_3 \cdot B_3 - R_X^{(T)} \cdot P = \\ = 2 \cdot 70 + 4 \cdot 21 + 4 \cdot 15 - 2 \cdot 105 = 74.$$

So the true rank of the number $X = (2 \| 4 \| 4) = 74$ is $R_X^{(T)} = 2$.

Conclusions. So, as the calculated rank $R_X^{(C)}$ of the number X is equal to the true rank $R_X^{(T)}$ of the number X, i.e. $R_X^{(C)} = R_X^{(T)} = 2$, then the calculated rank is determined correctly.

Let's determine the calculated rank $R_Y^{(C)}$ of the second summand $Y = (2 \| 3 \| 1) = 8$. First, as for the first summand X, let's reduce the number Y to the form $Y_n = (0 \| 0 \| 0)$, i.e. let's zero the number Y according to the first base $p_1 = 3$, adding the minimum ZC $z^{(1)}$ to the original number Y:

$$Y_1 = Y + z^{(1)} = (2 \| 3 \| 1) + (1 \| 1 \| 1) = (0 \| 4 \| 2).$$

Using expression (13) we determine the calculated rank of the number $Y_1 = (0 \| 4 \| 2)$:

$$R_{Y_1}^{(C)} = R_Y^{(C)} + R_{z^{(1)}} - \sum_{i=1}^{3}\left[\frac{y_i + z_j^{(1)}}{p_i}\right] \cdot e_i = R_Y^{(C)} + 1 - \\ -\left\{\left[\frac{y_1 + z_1^{(1)}}{p_1}\right] \cdot e_1 + \left[\frac{y_2 + z_2^{(1)}}{p_2}\right] \cdot e_2 + \left[\frac{y_3 + z_3^{(1)}}{p_3}\right] \cdot e_3\right\} = \\ = R_Y^{(C)} + 1 - \left\{\left[\frac{2+1}{3}\right] \cdot 2 + \left[\frac{3+1}{5}\right] \cdot 1 + \left[\frac{1+1}{7}\right] \cdot 1\right\} = \\ = R_Y^{(C)} + 1 - 1 \cdot 2 - 0 \cdot 1 - 0 \cdot 1 = R_Y^{(C)} + 1 - 2 = R_Y^{(C)} - 1.$$

So, since the residue $y_2 = 4$ has not been reset to zero, then let's add the value of the minimum ZC $z^{(2)} = (0 \| 3 \| 3)$:

$$Y_2 = Y_1 + z^{(2)} = (0 \| 4 \| 2) + (0 \| 3 \| 3) = (0 \| 2 \| 5).$$

$$R_{Y_2}^{(C)} = R_{Y_1}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3}\left[\frac{y_i + z_j^{(2)}}{p_i}\right] \cdot e_i = \left(R_Y^{(C)} - 1\right) + R_{z^{(2)}} - \\ -\left\{\left[\frac{y_1 + z_1^{(2)}}{p_1}\right] \cdot e_1 + \left[\frac{y_2 + z_2^{(2)}}{p_2}\right] \cdot e_2 + \left[\frac{y_3 + z_3^{(2)}}{p_3}\right] \cdot e_3\right\} = \\ = \left(R_Y^{(C)} - 1\right) + 1 - \left\{\left[\frac{0+0}{3}\right] \cdot 2 + \left[\frac{4+3}{5}\right] \cdot 1 + \left[\frac{2+3}{7}\right] \cdot 1\right\} = \\ = R_Y^{(C)} - 1 + 1 - 0 \cdot 2 - 1 \cdot 1 - 0 \cdot 2 = R_Y^{(C)} - 1.$$

After carrying out the stage of zeroing the residue modulo $p_2 = 5$, we obtain the value $y_2 = 2$. Thus, it is necessary to carry out one more time zeroing the residue $y_2 = 2$ of the number $Y_2 = (0 \| 2 \| 5)$ modulo $p_2 = 5$. Let's add the minimum ZC $z^{(2)}$ again:

$$Y_3 = Y_2 + z^{(2)} = (0 \| 2 \| 5) + (0 \| 3 \| 3) = (0 \| 0 \| 1).$$

$$R_{Y_3}^{(C)} = R_{Y_2}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3}\left[\frac{y_i + z_j^{(2)}}{p_i}\right] \cdot e_i = \\ \left(R_Y^{(C)} - 1\right) + R_{z^{(2)}} - \left\{\left[\frac{0+0}{3}\right] \cdot 2 + \left[\frac{2+3}{5}\right] \cdot 1 + \left[\frac{5+3}{7}\right] \cdot 1\right\} = \\ = R_Y^{(C)} - 1 + 1 - 0 \cdot 2 - 1 \cdot 1 - 1 \cdot 1 = R_Y^{(C)} - 2.$$

Let's reset the residue $y_3 = 1$ to zero according to the third module $p_3 = 7$ of the number $Y_3 = (0 \| 0 \| 1)$. Let's add the number Y_3 and minimum ZC $z^{(3)} = (0 \| 0 \| 1)$ and get:

$$Y_4 = Y_3 + z^{(3)} = (0 \| 0 \| 1) + (0 \| 0 \| 1) = (0 \| 0 \| 2).$$

$$R_{Y_4}^{(C)} = R_{Y_3}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3}\left[\frac{y_i + z_j^{(3)}}{p_i}\right] \cdot e_i = \left(R_Y^{(C)} - 2\right) + R_{z^{(3)}} - \\ -\left\{\left[\frac{y_1 + z_1^{(3)}}{p_1}\right] \cdot e_1 + \left[\frac{y_2 + z_2^{(3)}}{p_2}\right] \cdot e_2 + \left[\frac{y_3 + z_3^{(3)}}{p_3}\right] \cdot e_3\right\} = \\ = \left(R_Y^{(C)} - 2\right) + 0 - \left(\left[\frac{0+0}{3}\right] \cdot 2 + \left[\frac{0+0}{5}\right] \cdot 1 + \left[\frac{1+1}{7}\right] \cdot 1\right) = \\ = R_Y^{(C)} - 2 + 0 - 0 \cdot 2 - 0 \cdot 1 - 0 \cdot 1 = R_Y^{(C)} - 2.$$

After adding four more times with minimum ZC $z^{(3)}$ we get the number $Y_8 = (0 \| 0 \| 6)$. Add to this number the value of ZC $z^{(3)}$:

$$Y_n = Y_8 + z^{(3)} = (0 \| 0 \| 6) + (0 \| 0 \| 1) = (0 \| 0 \| 0).$$

Let's determine the calculate rank for the zeroable number $Y_n = (0 \| 0 \| 0)$:

$$R_{Y_n}^{(C)} = R_{Y_8}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3}\left[\frac{y_i + z_j^{(3)}}{p_i}\right] \cdot e_i = \\ \left(R_Y^{(C)} - 2\right) + R_{z^{(3)}} - \left\{\left[\frac{0+0}{3}\right] \cdot 2 + \left[\frac{0+0}{5}\right] \cdot 1 + \left[\frac{6+1}{7}\right] \cdot 1\right\} = \\ = R_Y^{(C)} - 2 + 0 - 0 \cdot 2 - 0 \cdot 2 - 1 \cdot 1 \\ = R_Y^{(C)} - 3.$$

According to expression (11) we have that:

$$R_{Y_n}^{(C)} = -1 \rightarrow R_Y^{(C)} - 3 = -1 \rightarrow R_Y^{(C)} = 2.$$

DOI 10.15588/1607-3274-2024-1-21

Thus, the calculated rank of the number $Y=(2\|3\|1)=8$ is $R_Y^{(C)}=2$.

Checking. In the PNS, the value of the second summand $Y=(2\|3\|1)=8$ is equal to the value:

$$Y_{PNS}=\sum_{i=1}^{3} y_i \cdot B_i = 2\cdot 70+3\cdot 21+1\cdot 15=$$
$$=218-R_Y^{(T)}\cdot 105=218-2\cdot 105=8.$$

So the true rank of the number $Y=(2\|3\|1)=8$ is $R_Y^{(T)}=2$.

Conclusions. Since the calculated rank of the number Y is equal to the true rank of the number Y, i.e. $R_Y^{(C)}=R_Y^{(T)}=2$, then the calculated rank is determined correctly.

Let's determine the sum of two numbers $X+Y$:

$$X+Y=(2\|4\|4)+(2\|3\|1)=(1\|2\|5).$$

According to expression (1), the calculated rank $R_{X+Y}^{(C)}$ of the sum of two numbers $X+Y$ is equal to:

$$R_{X+Y}^{(C)}=R_X^{(C)}+R_Y^{(C)}-\sum_{i=1}^{3}\left[\frac{x_i+y_i}{p_i}\right]\cdot e_i=$$
$$=R_X^{(C)}+R_Y^{(C)}-\left\{\left[\frac{2+2}{3}\right]\cdot 2+\left[\frac{4+3}{5}\right]\cdot 1+\left[\frac{4+1}{7}\right]\cdot 1\right\}=$$
$$=2+2-1\cdot 2-1\cdot 1-0\cdot 1=1.$$

Checking. $(X+Y)_{SRC}=(1\|2\|5)$ and $(X+Y)_{PNS}=$ $=1\cdot 70+2\cdot 21+5\cdot 15-R_{X+Y}^{(T)}\cdot P=187-1\cdot 105=82.$

Conclusion. The true rank $R_{X+Y}^{(T)}$ of the number $X+Y=(1\|2\|5)$ is equal to the calculated $R_{X+Y}^{(T)}=R_{X+Y}^{(C)}=1$. Therefore, there was no overflow when performing the addition operation.

Example 4. Check for overflow when adding two numbers $X=Y=(2\|4\|4)=74$.

Let's determine the sum of two numbers $X+Y$:

$$X+Y=(2\|4\|4)+(2\|4\|4)=(1\|3\|1).$$

Considering that in example 3, the calculated rank $R_X^{(C)}$ of the number $X=(2\|4\|4)=74$ was calculated, since the numbers X and Y are the same, its calculated ranks are also the same: $R_X^{(C)}=R_Y^{(C)}=2$.

According to expression (1), the calculated rank $R_{X+Y}^{(C)}$ of the sum $X+Y$ of two numbers in the SRC is equal to:

$$R_{X+Y}^{(C)}=R_X^{(C)}+R_Y^{(C)}-\sum_{i=1}^{3}\left[\frac{x_i+y_i}{p_i}\right]\cdot e_i=$$
$$=2+2-\left\{\left[\frac{2+2}{3}\right]\cdot 2+\left[\frac{4+4}{5}\right]\cdot 1+\left[\frac{4+4}{7}\right]\cdot 1\right\}=$$
$$=2+2-1\cdot 2-1\cdot 1-1\cdot 1=0.$$

Thus, the calculated rank of the sum of two numbers $X+Y$ is equal to the value $R_{X+Y}^{(C)}=0$.

Checking. $(X+Y)_{SRC}=(1\|3\|1)$ and $(X+Y)_{PNS}=$ $=1\cdot 70+3\cdot 21+1\cdot 15-R_{X+Y}^{(\mathrm{T})}\cdot P=148-R_{X+Y}^{(\mathrm{T})}\cdot 105=148-$ $-1\cdot 105=43$. Thus, $R_{X+Y}^{(\mathrm{T})}=1$.

Conclusion. It is obvious that the true rank $R_{X+Y}^{(T)}=1$ doesn't coincide with the calculated rank $R_{X+Y}^{(C)}=0$. The inequality $R_{X+Y}^{(C)}\neq R_{X+Y}^{(T)}$ of the rank values of the number $X+Y$ shows that there was an overflow during the operation. Therefore, the sum of two numbers $X+Y$ has the wrong value: 43, not 148.

6 DISCUSSION

When solving CS computational problems, it becomes necessary to take into account the overflow of the bit grid that occurs during data processing. Analysis of these processes showed the following. To solve the problem associated with overflow of the CS bit grid, there are various approaches. One of them is the use of a wider bit grid to represent the meanings of numbers, i.e. Data processing is carried out on computers with a relatively large bit grid [23]. This allows you to increase the range of values of processed numbers that can be represented in the CS without taking into account the consequences of the overflow factor. However, this requires more memory and can increase the time it takes to solve a calculation problem, which is especially critical for real-time CS [24, 25]. The MDBGO proposed in the article is intended for use in a CS that operates in the SRC. The properties of the SRC (independence, equality and low-bit residues, the totality of which determines the non-positional code structure) and their use in creating the structure of the CS determine the interpretation of the CS in the SRC as a set of individual low-bit computers [7, 26]. Each computer operates using a specific SRC module [27, 28]. In this case, eliminating the consequences of overflowing the bit grid is carried out without interrupting the computational process, i.e. during the operation of the CS, without stopping the calculations.

The reliability and significance of the results obtained are due to the following factors:

– the research was carried out using the modern proven mathematical apparatus of number theory and the basic theoretical principles of machine arithmetic in the residual classes;

– the consistency of the results obtained with both the scientific provisions of the general theory of constructing the structures of positional CSs, and the theoretical provisions of the creation of CSs operating in the SRC;

– the coincidence of some theoretical conclusions with existing modern provisions on the prospects for the development of real-time CS;

– the results of the analysis of the given specific examples of the use of MDBGO for various initial data of the SRC.

DOI 10.15588/1607-3274-2024-1-21

This problem, solved in the article, was directly or indirectly considered in the monographs: Aksushskyi I. Ya. and Yuditskyi D. I. "Machine arithmetic in residual classes" [1] and Torgashov V. A. "System of residual classes and reliability of digital computers" [8]. These monographs provide directions for further research in the direction of improving real-time CS structures in the SRC. In particular, some theoretical further research is presented.

CONCLUSIONS

The current scientific problem of using MDBGO in the CS operating in the SRC has been solved. The use of MDBGO to detect the fact of overflow of the CS bit grid is shown using specific examples of the implementation of the operation of adding two numbers in the SRC. The MDBGO considered in the article is based on the use of positional feature of a non-positional code of numbers in the SRC, namely on the calculation and use of the true and calculated ranks of numbers.

The scientific novelty of the results obtained lies in the fact that when implementing MDBGO, the procedure for determining the rank of a number is carried out directly in the process of performing the operation of adding two numbers, being an essential part of it. This circumstance makes it possible to reduce the time it takes to detect the fact of overflow of the CS bit grid in the SRC. In addition, a feature of the presented method for detecting the fact of overflow of the CS bit grid in the SRC is that MDBGO can simultaneously be used to organize the process of monitoring the operation of adding numbers modulo. This expands the functionality of the MDBGO.

Practical significance of the results. To confirm the practical feasibility of the procedure, examples are given of determining the overflow of the result of the operation of adding two numbers in the SRC. A set of examples is given of the specific implementation of the operation of overflowing the bit grid using the MDBGO method, while simultaneously implementing control of the addition of two numbers for a given SRC, which confirm the effectiveness of using the considered method.

Prospects for further research are as follows. In the SRC, using the basic properties of the class of residues, control of arithmetic operations can be carried out in the dynamics of the computational process, i.e. without stopping the calculation process. This makes it possible, firstly, to fully use the main property of the SRC – the high speed of the CS execution of arithmetic modular operations. Secondly, it is possible to reduce the amount of the CS equipment required to implement positional operations in the SRC. The research results obtained in the article are recommended for use in on-board digital computers of ballistic missiles, in the use of unmanned aerial vehicle computers and in the use of specialized computers for a wide class of various non-recoverable disposable aircraft operating in the SRC. The feasibility of further research in the field of using non-positional code structures in the SRC, in particular, expanding the area of practical use of MDBGO, is due to the fact that positive research results will make it possible to create ultra-fast and fault-tolerant real-time specialized CS.

REFERENCES

1. Akushskij I. Ya., Yudickij D. I. Mashinnaya arifmetika v ostatochnyx klassax. Moscow, Sovet. radio, 1968, 440 p.
2. Krasnobayev V., Yanko A., Hlushko A. Information Security of the National Economy Based on an Effective Data Control Method, *Journal of International Commerce, Economics and Policy*, 2023, No. 14(3), Article no. 2350021. DOI: 10.1142/S1793993323500217
3. Butt M. A., Ajmal Z., Khan Z. I. , Idrees M., Javed Y. An In-Depth Survey of Bypassing Buffer Overflow Mitigation Techniques, *Applied Sciences,* 2022, No. 12(13), P. 6702. DOI: 12. 6702. 10.3390/app12136702
4. 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, R*adio Electronics, Computer Science, Control,* 2022, No. (2), pp. 113–121. DOI: https://doi.org/10.15588/1607-3274-2022-2-11
5. Laktionov O., Lievi L., Tretia A., Movin M. Investigation of combined ensemble methods for diagnostics of the quality of interaction of human-machine systems, *Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu*, 2023, No. 4, pp. 138–143. DOI: https://doi.org/10.33271/nvngu/2023-4/138
6. Salnikov D., Karaman D., Krylova V. Highly reconfigurable soft-CPU based peripheral modules design, *Advanced Information Systems*, 2023, No. 7(2), pp. 92–97. DOI: https://doi.org/10.20998/2522-9052.2023.2.13
7. Yanko A., Koshman S., Krasnobayev V. Algorithms of data processing in the residual classes system, *International Scientific-Practical Conference: Problems of Infocommunications. Science and Technology (PIC S&T). Kharkov,* 2017, pp. 117–121. DOI: 10.1109/INFOCOMMST.2017.8246363
8. Torgashev V. A. Sistema ostatochnyx klassov i nadezh-nost' CVM. Moscow, Sovet. radio, 1973, 118 p.
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. Colbert I., Pappalardo A., Petri-Koenig J. A2Q: Accumulator-Aware Quantization with Guaranteed Overflow Avoidance Supplementary Material, *2023 IEEE/CVF International Conference on Computer Vision (ICCV). Paris,* France, 2023, pp. 16943–16952. DOI: 10.1109/ICCV51070.2023.0155
11. Bo L., Ruifeng Z., Jiangang L., Wenxin G., Yang L. Control on Abnormal Data Overflow of Distribution Network Management Platform, *Journal of Physics: Conference Series 1748*, 2021, No. 032064, pp. 1–5. DOI: 10.1088/1742-6596/1748/3/032064
12. Meakin R. L. Adaptive spatial partitioning and refinement for overset structured grids, *Computer Methods in Applied Mechanics and Engineering*, 2000, No. 189 (4), pp. 1077–1117. DOI: https://doi.org/10.1016/S0045-7825(99)00369-2
13. Hui H. W., Zhou C. C., Xu S. G., Lin F. H. A Novel Secure Data Transmission Scheme in Industrial Internet of Things, *China Communications*, 2020, No. 17 (1), pp. 73–88. DOI: 10.23919/JCC.2020.01.006
14. Chiamwongpaet S., Piromsopa K. Boundary Bit: Architectural Bound Checking for Buffer-Overflow Protection, *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, 2020, No. 14, pp. 162–173. DOI:10.37936/ecti-cit.2020142.212338
15. Tsukada M., Matsutani H. An Overflow/Underflow-Free Fixed-Point Bit-Width Optimization Method for OS-ELM Digital Circuit, *IEICE Trans. Fundamentals, Vol. E105–A,* No. 3, 2022, pp. 437–447. DOI: 10.1587/transfun.2021VLP0017
16. Xie H., Song Y., Cai L., Li M. Overflow Aware Quantization: Accelerating Neural Network Inference by Low-bit Multiply-Accumulate Operations, *29th International Joint Conference on*

DOI 10.15588/1607-3274-2024-1-21

Artificial Intelligence. Japan, 2020, pp. 868–875. DOI: https://doi.org/10.24963/ijcai.2020/121

17. Lee S., Shin D. J. Overflow-Detectable Floating-Point Fully Homomorphic Encryption, *IEEE Access*, 2024, pp. 1–21. DOI: 10.1109/ACCESS.2024.3351738
18. Li T., Wang Y., Li R., Li K. A bidirectional overflow digital correction algorithm with a single bit redundancy used in the pipeline A/D converters, *2009 IEEE 8th International Conference on ASIC.* Changsha, Hunan, 2009, pp. 238–241. DOI: 10.1109/ASICON.2009.5351482
19. Cha Y., Cho G., Choi H., Song H. N bit result integer multiplier with overflow detector, *Electronics Letters,* 2001, No.(37), pp. 940–942. DOI: 10.1049/el:20010633.
20. Ivohin E. V. Gavrylenko V. V., Ivohina K. E. On the recursive algorithm for solving the traveling salesman problem on the basis of the data flow optimization method, *Radio Electronics, Computer Science, Control*, 2023, No. (3), pp. 141–147. DOI: https://doi.org/10.15588/1607-3274-2023-3-14
21. 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
22. Krasnobayev V. A., Yanko A. S., Kovalchuk D. M. Methods for tabular implementation of arithmetic operations of the residues of two numbers represented in the system of residual classes, *Radio Electronics, Computer Science, Control*, 2022, No. 4, pp. 18–27. DOI: https://doi.org/10.15588/1607-3274-2022-4-2
23. Subbotin S. A. Data clustering based on inductive learning of neuro-fuzzy network with distance hashing, *Radio Electronics, Computer Science, Control*, 2022, No. 4, pp. 71–85. DOI: https://doi.org/10.15588/1607-3274-2022-4-6
24. Krasnobayev V. Koshman S., Yanko A., Martynenko A. Method of Error Control of the Information Presented in the Modular Number System, *International Scientific-Practical Conference: Problems of Infocommunications Science and Technology (PIC S&T)*. Kharkov, 2018, pp. 39–42. DOI: 10.1109/INFOCOMMST.2018.8632049
25. Chao Huang, Peterson D., Rauch H., Teague J., Fraser D. Implementation of a fast digital processor using the residue number system, *IEEE Transactions on Circuits and Systems,* 1981, January, Vol. 28, № 1, pp. 32–38. DOI: 10.1109/TCS.1981.1084905
26. Szabo N. S., Tanaka R. I. Residue Arithmetic and Its Applications to Computer Technology. New York, McGraw-Hill, 1967, 236 p.
27. 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
28. Salnikov D., Karaman D., Krylova V. Highly reconfigurable soft-CPU based peripheral modules design, *Advanced Information Systems,* 2023, No. 7(2), pp. 92–97. DOI: https://doi.org/10.20998/2522-9052.2023.2.13

Received 19.01.2024.
Accepted 28.02.2023.

УДК 004.222.5:681.142.01

МЕТОД ВИЗНАЧЕННЯ ПЕРЕПОВНЕННЯ РОЗРЯДНОЇ СІТКИ КОМП'ЮТЕРНОЇ СИСТЕМИ, ЩО ФУНКЦІОНУЄ В СИСТЕМІ ЗАЛИШКОВИХ КЛАСІВ

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.

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

Нікольський С. Б. – канд. техн. наук, доцент, доцент кафедри інформатики комунального закладу «Харківської гуманітарно-педагогічної академії» Харківської обласної ради, Харків, Україна.

Крук О. О. – аспірант кафедри автоматики, електроніки та телекомунікацій Національного університету «Полтавська політехніка імені Юрія Кондратюка», Полтава, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто метод визначення переповнення розрядної сітки, а також комплекс прикладів практичного застосування процедури ідентифікації переповнення розрядної сітки комп'ютерної системи, що функціонує в непозиційній системі числення в залишкових класах. Об'єктом дослідження є процес обробки даних, представлених у системі залишкових класів. Мета роботи – розглянути та проаналізувати приклади визначення переповнення розрядної сітки комп'ютерної системи при реалізації операції додавання двох чисел у системі залишкових класів на основі застосування методу визначення переповнення розрядної сітки, заснованого на використанні поняття рангу числа.

Метод. Специфіка функціонування комп'ютерної системи у системі залишкових класів вимагає виконання як модульних операцій, так й реалізації додатково, так званих, немодульних операцій. До немодульних операцій належить операція визначення переповнення розрядної сітки комп'ютерної системи у системі залишкових класів. У непозиційній системі числення в залишкових класах реалізація процесу виявлення переповнення розрядної сітки комп'ютерної системи є важко реалізованим завданням. Розглянутий у статті метод визначення переповнення розрядної сітки ґрунтується на використанні позиційних ознак непозиційного коду чисел у системі залишкових класів, а саме істинного та розрахункового рангів числа. Досліджено процес визначення переповнення розрядної сітки результату операції додавання двох чисел у системі залишкових класів, оскільки саме виконання арифметичної операції додавання є основною, базовою операцією комп'ютерної системи.

Результати. Наведено приклади використання методу визначення переповнення результату операції додавання двох чисел у системі залишкових класів, в основу якого покладено модульні операції визначення розрахункового та істинного рангів безпосередньо доданків та рангу суми двох доданків. Аналіз отриманих результатів показав практичну застосовність розглянутого методу.

Висновки. Основним перевагою представленого методу є те, що визначення переповнення розрядної сітки можна здійснювати у динаміці обчислювального процесу комп'ютерної системи, тобто без зупинки розв'язання задачі. Ця обставина дозволяє знизити непродуктивні витрати комп'ютерної системи в системі залишкових класів. Крім цього, цей метод можна використовувати для контролю операції додавання двох чисел в системі залишкових класів. Це підвищує достовірність отримання істинного результату операції додавання двох чисел в системі залишкових класів.

DOI 10.15588/1607-3274-2024-1-21

КЛЮЧОВІ СЛОВА: арифметична операція модульного додавання, переповнення розрядної сітки, операція порівняння, комп'ютерна система, непозиційний код, ранг числа, система залишкових класів, процедура нулевізації.

ЛІТЕРАТУРА

1. Акушский И.Я. Машинная арифметика в остаточных классах / И. Я. Акушский, Д. И. Юдицкий. – Москва : Совет. радио, 1968. – 440 с.
2. Krasnobayev V. Information Security of the National Economy Based on an Effective Data Control Method / V. Krasnobayev, A. Yanko, A. Hlushko // Journal of International Commerce, Economics and Policy. – 2023. – No. 14(3). – Article no. 2350021. DOI: 10.1142/S1793993323500217
3. Butt M. A. An In-Depth Survey of Bypassing Buffer Overflow Mitigation Techniques / [M. A. Butt, Z. Ajmal, Z. I. Khan et al.] // Applied Sciences. – 2022. – No. 12(13), P. 6702. DOI: 12. 6702. 10.3390/app12136702
4. Gavrylenko S. Y. Development of method for identification the computer system state based on the decision tree with multi-dimensional 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
5. Investigation of combined ensemble methods for diagnostics of the quality of interaction of human-machine systems / [O. Laktionov, L. Lievi, A.Tretia, M. Movin] // Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu. – 2023. – No. (4). – P. 138–143. DOI: https://doi.org/10.33271/nvngu/2023-4/138
6. Salnikov D. Highly reconfigurable soft-CPU based peripheral modules design / D. Salnikov, D. Karaman, V. Krylova // Advanced Information Systems, 7(2). – 2023. – P. 92–97. DOI: https://doi.org/10.20998/2522-9052.2023.2.13
7. Yanko A. Algorithms of data processing in the residual classes system / A. Yanko, S. Koshman, V. Krasnobayev // International Scientific-Practical Conference: Problems of Infocommunications. Science and Technology (PIC S&T), Kharkov. – 2017. – P. 117–121. DOI: 10.1109/INFOCOMMST.2017.8246363
8. Торгашев В. А. Система остаточных классов и надежность ЦВМ / В. А. Торгашев. – Москва : Совет. радио, 1973. – 118 с.
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. Colbert I. A2Q: Accumulator-Aware Quantization with Guaranteed Overflow Avoidance Supplementary Material / I. Colbert, A. Pappalardo, J. Petri-Koenig // 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France. – 2023. – P. 16943–16952. DOI: 10.1109/ICCV51070.2023.0155
11. Control on Abnormal Data Overflow of Distribution Network Management Platform / [L. Bo, Z. Ruifeng, L. Jiangang et al.] // Journal of Physics: Conference Series 1748. –2021. – No. 032064. – P. 1–5. DOI: 10.1088/1742-6596/1748/3/032064
12. Meakin R. L. Adaptive spatial partitioning and refinement for overset structured grids / R. L. Meakin // Computer Methods in Applied Mechanics and Engineering. – 2000. – No. 189 (4). – P. 1077–1117. DOI: https://doi.org/10.1016/S0045-7825(99)00369-2
13. A Novel Secure Data Transmission Scheme in Industrial Internet of Things / [H. W. Hui, C. C. Zhou, S. G. Xu, F. H. Lin] // China Communications. – 2020. – No. 17 (1). – P. 73–88. DOI: 10.23919/JCC.2020.01.006
14. Chiamwongpaet S. Boundary Bit: Architectural Bound Checking for Buffer-Overflow Protection / S. Chiamwongpaet, K. Piromsopa // ECTI Transactions on Computer and Information Technology (ECTI-CIT). – 2020. – No. 14. – P. 162–173. DOI:10.37936/ecti-cit.2020142.212338
15. Tsukada M. An Overflow/Underflow-Free Fixed-Point Bit-Width Optimization Method for OS-ELM Digital Circuit / M. Tsukada, H. Matsutani // IEICE Trans. Fundamentals, 2022. – Vol. E105–A, No. 3. – P. 437–447. DOI: 10.1587/transfun.2021VLP0017
16. Overflow Aware Quantization: Accelerating Neural Network Inference by Low-bit Multiply-Accumulate Operations / [H. Xie, Y. Song, L. Cai, M. Li] // 29th International Joint Conference on Artificial Intelligence, Japan, 2020. – P. 868–875. DOI: https://doi.org/10.24963/ijcai.2020/121
17. Lee S. Overflow-Detectable Floating-Point Fully Homomorphic Encryption / S. Lee, D. J. Shin // IEEE Access. – 2024. –P. 1–21. DOI: 10.1109/ACCESS.2024.3351738
18. A bidirectional overflow digital correction algorithm with a single bit redundancy used in the pipeline A/D converters / [T. Li, Y. Wang, R. Li, K. Li] // 2009 IEEE 8th International Conference on ASIC, Changsha, Hunan, 2009. –P. 238–241. DOI: 10.1109/ASICON.2009.5351482
19. N bit result integer multiplier with overflow detector / [Y. Cha, G. Cho, H. Choi, H. Song] // Electronics Letters, 2001. – No.(37). – P. 940–942. DOI: 10.1049/el:20010633.
20. Ivohin E. V. On the recursive algorithm for solving the traveling salesman problem on the basis of the data flow optimization method / E. V. Ivohin, V. V. Gavrylenko, K. E. Ivohina // Radio Electronics, Computer Science, Control. – 2023. – No. (3). – P. 141–147. DOI: https://doi.org/10.15588/1607-3274-2023-3-14
21. 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
22. Krasnobayev V. A. Methods for tabular implementation of arithmetic operations of the residues of two numbers represented in the system of residual classes / V. A. Krasnobayev, A. S. Yanko, D. M. Kovalchuk // Radio Electronics, Computer Science, Control. – 2022. – No.(4). – P. 18–27. DOI: https://doi.org/10.15588/1607-3274-2022-4-2
23. Subbotin S. A. Data clustering based on inductive learning of neuro-fuzzy network with distance hashing / S. A. Subbotin // Radio Electronics, Computer Science, Control. –2022. – No. (4). – P. 71–85. DOI: https://doi.org/10.15588/1607-3274-2022-4-6
24. Method of Error Control of the Information Presented in the Modular Number System / [V. Krasnobayev, S. Koshman, A. Yanko, A. Martynenko] // International Scientific-Practical Conference: Problems of Infocommunications Science and Technology (PIC S&T). – Kharkov, 2018. – P. 39–42. DOI: 10.1109/INFOCOMMST.2018.8632049
25. Chao Huang Implementation of a fast digital processor using the residue number system / [Chao Huang, D. Peterson, H. Rauch et al.] // IEEE Transactions on Circuits and Systems. 1981. – January, Vol. 28, № 1. – P. 32–38. DOI: 10.1109/TCS.1981.1084905
26. Szabo N. S. Residue Arithmetic and Its Applications to Computer Technology / N. S. Szabo, R. I. Tanaka. – New York : McGraw-Hill, 1967. – 236 p.
27. 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
28. Salnikov D. Highly reconfigurable soft-CPU based peripheral modules design / D. Salnikov, D. Karaman, V. Krylova // Advanced Information Systems. – 2023. – No. 7(2). P. 92–97. DOI: https://doi.org/10.20998/2522-9052.2023.2.13

DOI 10.15588/1607-3274-2024-1-21