Adv Syst Sci Appl 2023; **04**:31–39
Published online at https://ijassa.ipu.ru.

Study of a Redundant Residue Number System for Single Error Correction

Viktor Kuchukov[1]*, Mikhail Babenko[2], Safwat Al-Galda[3]

[1] *North-Caucasus Center for Mathematical Research, North-Caucasus Federal University, Stavropol, Russia*
[2] *North-Caucasus Federal University, Stavropol, Russia*
[3] *University of Misan, Amarah, Iraq*

Abstract: One approach to reliability is to use a redundant residue number system. In general, two redundant moduli are required to detect and correct a single error. This paper considers an approach to error correction using a single redundant modulo, which allows a significant reduction in the hardware used, but at a significant cost in computational speed. The use of an approximate method based on the Chinese remainder theorem allows the speed of computation to be increased by eliminating the computationally complex operation of taking the remainder from the division by the range of the residue number system. A method based on the approximate method with one redundant modulo is proposed. Modelling of the considered methods on ASICs in RTL and physical synthesis environment Cadence Genus Synthesis Solution is carried out.

Keywords: residue number system, error correction, Chinese remainder theorem, application-specific integrated circuit

1. INTRODUCTION

The problem of reliability and guaranteed validity of the results obtained is relevant for remote control systems where autonomy and fault tolerance are important, e.g. on railways and oil and gas pipelines. It is known that for data transmission channels the probability of single errors significantly exceeds the probability of errors of higher multiplicity (generalised binomial law of error distribution), and for digital devices the law of error distribution is close to uniform [1, 2]. Most correction codes, especially block binary codes, which include parity check codes, have two groups of digits — information and control digits. The information group contains digits that represent the numerical value of the coded value, and the control group contains digits that are redundant and additionally introduced for the purpose of detecting and correcting possible distortions [3].

Since there is no possibility of composing the control part of the result by the control parts of the arithmetic operation components, the possibility of controlling the correctness of arithmetic operations is excluded. This non-arithmetic character of special position codes prevents their use in calculating machines, for which the control of arithmetic operations is of particular importance. One of the examples of arithmetic codes are AN codes [4, 5], but these codes allow only addition and subtraction operations. Another approach to the construction of arithmetic codes is the residue number system (RNS).

Further, the paper is organized as follows. Section 2 considers a redundant residue number system and shows the basic properties of error correction in RNS. Section 3 considers the

*Corresponding author: vkuchukov@ncfu.ru residue number system with one redundant modulo for single error correction. Section 4 shows the application of the approximate method based on the Chinese remainder theorem for error correction in RNS. Section 5 is devoted to the application of the approximate method to RNS with one redundant modulo. Section 6 presents the modelling of the considered methods.

2. REDUNDANT RESIDUE NUMBER SYSTEM

One of the most promising representations of numbers in parallel processing is the use of non-positional number systems, such as the residue number system.

If a set of positive integers $p_1, p_2, \ldots, p_n$, called moduli of the system, is given, then a residue number system is a system in which a positive integer is represented as a set of residues on the chosen moduli $X = (x_1, x_2, \ldots, x_n)$, where $x_i = X \bmod p_i$ for $i = 1, 2, \ldots, n$ [3]. If the moduli satisfy the condition $p_1 < p_2 < \ldots < p_n$, then the system is called ordered.

It is known from number theory that if the moduli p_i are mutually prime, then the representation of the number $X = (x_1, x_2, \ldots, x_n) < P = p_1 \cdot p_2 \cdot \ldots \cdot p_n$ is singular, where P is the dynamic range of the number representation. The peculiarity of the residue number system is the possibility of performing addition, subtraction and multiplication operations in parallel and independently for each of the modulo [6].

If we add to the residue number system with moduli $\{p_1, p_2, \ldots, p_n\}$ and dynamic range $P = \prod_{i=1}^{n} p_i$, which is often called the working range, an additional modulo $p_{n+1} > p_i$, $i = \overline{1, n}$, then the full range of the system is $\overline{P} = P \cdot p_{n+1}$. In the case of operations on numbers in the range $[0, P)$, if the result of the operation is less than P, it is correct, otherwise it is incorrect. Thus, the introduction of the redundant modulo p_{n+1} allows us to detect calculation errors. The introduction of constraints on the redundant modulo or the introduction of additional moduli allows not only the detection but also the correction of errors [7, 8].

Let's take as an example the RNS $\{3, 5, 7, 8\}$, where $p_4 = 8$ is the control modulo. Then the working range $P = 105$, the full range $\overline{P} = 840$. Take the number $X = (1, 2, 3, 4) = 52$, since $X < P$, there is no error. Suppose there is an error on the second modulo and the number $\overline{X} = (1, 3, 3, 4) = 388$, and since $\overline{X} > P$, the error can be detected.

One of the methods of locating errors in modular code is the method of projections. The projection X_i of a number $X = (x_1, x_2, \ldots, x_{n+1})$ onto the modulo p_i is the number obtained by crossing out the digit x_i in the representation of X.

If in an ordered residue number system the projection X_i of a number $X = (x_1, x_2, \ldots, x_i, \ldots, x_n, x_{n+1})$ on the modulo of p_i satisfies the condition

$$X_i > \frac{\overline{P}}{p_{n+1}},$$

then the digit x_i is correct if only one error is possible [3]. The introduction of only one control modulo generally fails to locate the error. To correct the error, we can use a method based on the Chinese remainder theorem (CRT), by which the number X can be obtained from the formula

$$X = \left| \sum_{i=1}^{n} P_i \cdot x_i \cdot \left| P_i^{-1} \right|_{p_i} \right|_P, \tag{2.1}$$

where P is the dynamic range, $P_i = P/p_i$, $\left| P_i^{-1} \right|_{p_i}$ is the multiplicative inversion of P_i modulo p_i, and the operator $|X|_{p_i}$ denotes the remainder of the division of X by p_i, i.e. $X \bmod p_i$ [9].

Consider the above number $\overline{X} = (1, 3, 3, 4) = 388$, for which the projections are $X_1 = (3, 3, 4) = 108$, $X_2 = (1, 3, 4) = 52$, $X_3 = (1, 3, 4) = 28$, $X_4 = (1, 3, 3) = 73$. We can see that the projections X_2, X_3, X_4 are within the working range and the error cannot be corrected.

Let's take a RNS with two redundant moduli $\{3, 5, 7, 11, 13\}$ with working range $P = 105$ and full range $\overline{P} = 15015$. Let's introduce an error in the number $X = (1, 2, 3, 8, 0) = 52$ on the second modulo $\overline{X} = (1, 3, 3, 8, 0) = 6058$. Then the projections $X_1 = 1053$, $X_2 = 52$, $X_3 = 1768$, $X_4 = 598$, $X_5 = 283$. From here we can see that only the projection X_2 falls within the working range and then the remainder of $X_2 = 52$ modulo $p_2 = 5$ is 2 and the corrected number $X = (1, 2, 3, 8, 0)$.

3. CORRECTION OF AN ERROR WITH ONE REDUNDANT MODULO

The paper [10] proposes an approach to error correction with one redundant modulo. Given a residue number system with a control modulo $p_1 < p_2 < p_3 < \ldots < p_n < p_{n+1}$, it is assumed that the control modulo is reliable and cannot contain an error. If $p_{n+1} > p_n \cdot p_{n-1}$, then algorithm 1 is executed.

Algorithm 1. Number recovery based on the method of projections and CRT

```
Require: $X' = \left(x'_1, x'_2, \ldots, x'_n, x'_{n+1}\right)$
Ensure: $X$
    Data in memory: $\{p_1, p_2, \ldots, p_{n+1}\}$, $P = \prod_{i=1}^{n} p_i$, $\overline{P} = p_n \cdot P$
    $w_i = \left|\overline{P}_i^{-1}\right|_{p_i} \cdot \overline{P}_i$, where $\overline{P}_i = \overline{P}/p_i$, for all $i \in [1, n+1]$
    $w_{i,j} = \left|\overline{P}_{i,j}^{-1}\right|_{p_j} \cdot \overline{P}_{i,j}$, where $\overline{P}_{i,j} = \overline{P}_i/p_j$, for all $i \in [1, n+1]$ and $j \neq i$
$S = 0$
for $i = 1$ to $n + 1$ do
    $S = S + x'_i \cdot w_i$
end for
$S = S \bmod \overline{P}$
if $S < P$ then
    $X = S$
    return $X$
else
    for $i = 1$ to $n + 1$ do
        $S_i = 0$
        for $j = 1$ to $n + 1$ do
            if $i \neq j$ then
                $S_i = S_i + x'_i \cdot w_{i,j}$
            end if
        end for
        $X = S_i \bmod \overline{P}_i$
        if $X < P$ then
            return $X$
        end if
    end for
end if
```

This algorithm allows a working modulo error to be corrected. However, the error in the control modulo is not always corrected exactly. Let's take the RNS $\{3, 5, 7, 37\}$ and the number $X = (1, 2, 3, 15) = 52$ and introduce the error on the control modulo $\overline{X} = (1, 2, 3, 17) = 1312$. Algorithm 1 gives the number $X' = (2, 2, 3, 17) = 17$.

However, the number of projections required is reduced if the reliability of the calculations on the control modulo is ensured.

4. APPROXIMATE METHOD BASED ON THE CHINESE REMAINDER THEOREM

Formula (2.1) and Algorithm 1 use the operation of finding the remainder of the division by a large modulo P. This operation is computationally expensive. One way to improve the efficiency of this operation is to use an approximate method based on CRT.

To detect and correct a single error in the RNS, we consider the addition of two redundant moduli p_{n+1} and p_{n+2}. In [11, 12] a fractional approximate representation of numbers based on CRT is proposed. Let us divide (2.1) by $\overline{P}$ and we get

$$\frac{X}{\overline{P}} = \left| \sum_{i=1}^{n+2} x_i \cdot \frac{\left|\overline{P}_i^{-1}\right|_{p_i}}{p_i} \right|_1 = \left| \sum_{i=1}^{n+2} x_i \cdot k_i \right|_1, \tag{4.2}$$

where $k_i = \frac{\left|\overline{P}_i^{-1}\right|_{p_i}}{p_i}$ are constants of the chosen system, and the operation $|x|_1$ means taking the fractional part of the number x. In this case, the value of the expression (4.2) is in the interval $[0, 1)$. Taking the fractional part of a number is much simpler than finding the remainder of the division by the full range, but in hardware implementation the coefficients k_i can rarely be represented as finite fractions and there is a question of rounding accuracy.

To solve this problem, the fractional coefficients k_i are multiplied by 2^N, where N is the number of binary digits after the decimal point, providing the necessary level of accuracy of the calculations, each number obtained is rounded up to an integer, and then all calculations are performed modulo 2^N. Finding the remainder by this modulo is solved in hardware by truncation, which is a trivial task. For the calculations we can use the estimation proposed in [13]:

$$N = \left\lceil \log_2 \left(\overline{P} \cdot \sum_{i=1}^{n+2} (p_i - 1) \right) \right\rceil .$$

In this case, the working range P is represented in the redundant RNS and it is obvious that $P = (0, \ldots, 0, \pi_{n+1}, \pi_{n+2})$, where $\pi_{n+1} = P \bmod p_{n+1}$, $\pi_{n+2} = P \bmod p_{n+2}$. Thus, when calculating projections, it is necessary to find relative values for both $\overline{X}$ and P.

Let's look at a similar example in RNS with two redundant moduli $\{3, 5, 7, 11, 13\}$ and the number $\overline{X} = (1, 3, 3, 8, 0) = 6058$. According to formula (4.2), the coefficients k_i are $k_1 = 1/3$, $k_2 = 2/5$, $k_3 = 5/7$, $k_4 = 1/11$, $k_5 = 6/13$, rounded to the nearest $N = 19$. To reduce the record, we will use fractional representation.

Then the relative value for $\overline{X}$ is $466/1155$ and for P is $1/143$. Since $466/1155 > 1/143$, the number $\overline{X}$ contains an error.

For projection modulo p_1, the coefficients k_i are $k_1 = 1/5$, $k_2 = 1/7$, $k_3 = 3/11$, $k_4 = 5/13$. Then the relative value for $\overline{X}_1$ is $81/385$ and for P is $3/143$. Since $81/385 > 3/143$, the number $\overline{X}_1$ contains an error. For projection modulo p_2, the coefficients k_i are $k_1 = 2/3$, $k_2 = 4/7$, $k_3 = 5/11$, $k_4 = 4/13$. Then the relative value for $\overline{X}_2$ is $4/231$ and for P is $5/143$. Since $4/231 < 5/143$, the number $\overline{X}_2$ is error free. The other projections show that the number contains an error.

To recover the number we need to multiply the value of $4/231$ by the full range of the projection, i.e. by $\overline{P}/p_2 = 3003$, we get 52, which corresponds to the number without error.

5. APPLICATION OF THE APPROXIMATE METHOD TO ERROR CORRECTION WITH ONE REDUNDANT MODULO

Let us consider the application of the approximate method to Algorithm 1. The disadvantage of this algorithm is the need to find the remainder by the large modulo. Let us introduce Algorithm 2, which allows us to correct the working modulo error in RNS with one redundant modulo $p_{n+1} > p_n \cdot p_{n-1}$.

Algorithm 2. Number recovery based on the projection method and approximate CRT

```
Require: $X' = \left(x'_1, x'_2, \ldots, x'_n, x'_{n+1}\right)$
Ensure: $X$
    Data in memory: $\{p_1, p_2, \ldots, p_{n+1}\}$, $P = \prod_{i=1}^{n} p_i$,
    $\overline{P} = p_n \cdot P$, $\pi = P \bmod p_{n+1}$
    $\overline{P}_i = \overline{P}/p_i$, $w_i = \frac{\left|\overline{P}_i^{-1}\right|_{p_i}}{p_i}$, $i \in [1, n+1]$
    $\overline{P}_{i,j} = \overline{P}_i/p_j$, $w_{i,j} = \frac{\left|\overline{P}_{i,j}^{-1}\right|_{p_j}}{p_j}$, $i \in [1, n]$, $j \in [1, n+1]$, $i \neq j$
X = $\left|\sum_{i=1}^{n+1} {x'}_i \cdot w_i\right|_1$
$W = \left|\pi \cdot w_{n+1}\right|_1$
if $X < W$ then
    return $X \cdot \overline{P}$
else
    for $i = 1$ to $n$ do
        $X_i = \left|\sum_{j=1, j\neq i}^{n+1} x'_j \cdot w_{i,j}\right|_1$
        $W_i = \left|\pi \cdot w_{i,n+1}\right|_1$
        if $X_i < W_i$ then
            return $X_i \cdot \overline{P}_i$
        end if
    end for
end if
```

Let's look at an example of how Algorithm 2 works for RNS $\{3, 5, 7, 37\}$ and the number $\overline{X} = (1, 2, 3, 17) = 1312$.

The data stored in memory:

$$
\begin{aligned}
&P = 105, \overline{P} = 3885, \pi = 31,\\
&P_1 = 1295, P_2 = 777, P_3 = 555, P_4 = 105,\\
&P_{1,2} = P_{2,1} = 259, P_{1,3} = P_{3,1} = 185, P_{1,4} = P_{4,1} = 35, P_{2,3} = P_{3,2} = 111,\\
&P_{2,4} = P_{4,2} = 21, P_{3,4} = P_{4,3} = 15,
\end{aligned}
$$

$$
\begin{aligned}
w_1 &= \frac{2}{3}, w_2 = \frac{3}{5}, w_3 = \frac{4}{7}, w_4 = \frac{6}{37},\\
w_{1,2} &= \frac{4}{5}, w_{1,3} = \frac{5}{7}, w_{1,4} = \frac{18}{37},\\
w_{2,1} &= \frac{1}{3}, w_{2,3} = \frac{6}{7}, w_{2,4} = \frac{30}{37},\\
w_{3,1} &= \frac{2}{3}, w_{3,2} = \frac{1}{5}, w_{3,4} = \frac{5}{37}.
\end{aligned}
$$

Then

$$X = \left|1 \cdot \frac{2}{3} + 3 \cdot \frac{3}{5} + 3 \cdot \frac{4}{7} + 15 \cdot \frac{6}{37}\right|_1 = \frac{2383}{3885},\ W = \left|31 \cdot \frac{6}{37}\right|_1 = \frac{1}{37}.$$

Since $2383/3885 > 1/37$, the number contains an error. Let's construct the projections.

Then the first projection

$$X_1 = \left|3 \cdot \frac{4}{5} + 3 \cdot \frac{5}{7} + 15 \cdot \frac{18}{37}\right|_1 = \frac{1088}{1295},\ W_1 = \left|31 \cdot \frac{18}{37}\right|_1 = \frac{3}{37}.$$

Since $1088/3885 > 3/37$, the number contains an error. The second projection

$$X_2 = \left|1 \cdot \frac{1}{3} + 3 \cdot \frac{6}{7} + 15 \cdot \frac{30}{37}\right|_1 = \frac{52}{777},\ W_2 = \left|31 \cdot \frac{30}{37}\right|_1 = \frac{5}{37}.$$

Since $52/777 < 5/37$, the number does not contain an error. The third projection

$$X_3 = \left|1 \cdot \frac{2}{3} + 3 \cdot \frac{1}{5} + 15 \cdot \frac{5}{37}\right|_1 = \frac{373}{555},\ W_3 = \left|31 \cdot \frac{5}{37}\right|_1 = \frac{7}{37}.$$

Since $373/555 > 7/37$, the number contains an error.

So the error was in the second modulo. To recover the number, it is necessary to multiply $52/777$ by $P_2 = 777$, so the correct number is 52. This approach can also contain an error only on working moduli.

6. MODELLING

Single error correction simulations were performed on an ASIC in the RTL and physical synthesis Cadence Genus Synthesis Solution environment using the osu018_stdcells library. The metrics measured were the time for the signal to travel through the circuit (picoseconds, ps), the power required (watts, W) and the area used (square microns, μm^2).

RNS sets with one and two redundant moduli with 4-6 working moduli covering the 8, 16, 24 and 32 bit ranges shown in Table 6.1 are selected for the simulation.

Tables 6.2–6.4 show the simulation results for the signal propagation time through the circuit (picoseconds, ps), the area used (square microns, μm^2) and the power required (watts, W). Algorithm 1 with one redundant modulo, denoted as `I` in the tables, an approximate method based on CRT with two redundant moduli, denoted as `II`, an approximate method based on CRT with one redundant modulo, denoted as `III`, are chosen for the simulations.

Tables 6.2–6.4 show that the approximate method with two moduli is on average 0.43% faster, uses 16.96% more area and has a 25.44% higher power consumption compared to the approximate method with a one redundant modulo. The single redundant modulo method is on average 180% slower, uses 23% more power but 69% less area.

Table 6.1. Sets of moduli for modelling

	Number of working moduli	Working range size, bits			
		8	16	24	32
One redundant modulo	4	{3, 5, 7, 11, 79}	{13, 17, 19, 23, 439}	{59, 61, 67, 71, 4759}	{251, 257, 263, 269, 70753}
	5	{2, 3, 5, 7, 11, 79}	{5, 7, 11, 13, 17, 223}	{23, 29, 31, 37, 41, 1523}	{79, 83, 89, 97, 101, 9803}
	6	{2, 3, 5, 7, 11, 13, 149}	{3, 5, 7, 11, 13, 17, 223}	{11, 13, 17, 19, 23, 29, 673}	{31, 37, 41, 43, 47, 53, 2503}
Two redundant moduli	4	{3, 5, 7, 11, 13, 17}	{13, 17, 19, 23, 29, 31}	{59, 61, 67, 71, 73, 79}	{251, 257, 263, 269, 271, 277}
	5	{2, 3, 5, 7, 11, 13, 17}	{5, 7, 11, 13, 17, 19, 23}	{23, 29, 31, 37, 41, 43, 47}	{79, 83, 89, 97, 101, 103, 107}
	6	{2, 3, 5, 7, 11, 13, 17, 19}	{3, 5, 7, 11, 13, 17, 19, 23}	{11, 13, 17, 19, 23, 29, 31, 37}	{31, 37, 41, 43, 47, 53, 59, 61}

Table 6.2. Results of the simulation of the signal propagation time through the circuit, ps

Number of working moduli	Algorithm	Working range size, bits			
		8	16	24	32
4	I	25097	39429	84673	119623
	II	11369	15876	21509	31825
	III	11476	15707	24436	29211
5	I	22903	36698	76512	111360
	II	12424	15646	23134	27172
	III	12374	16354	22028	27055
6	I	29077	41193	66569	66711
	III	15867	17340	22252	27402
	IV	15542	18717	21620	26845

Table 6.3. Results of the simulation of the used area, μm^2

Number of working moduli	Algorithm	Working range size, bits			
		8	16	24	32
4	I	60759	132332	319916	538922
	II	186419	393166	816822	1470043
	III	148011	314708	684626	1206436
5	I	62136	131767	287343	493900
	II	258575	422888	877680	1448269
	III	200183	347296	715083	1329741
6	I	94223	155115	282315	399128
	II	444720	566191	916218	1393492
	III	343388	473257	783562	1281555

Table 6.4. Results of the simulation of the required power, W

Number of working moduli	Algorithm	Working range size, bits			
		8	16	24	32
4	I	4,82E-01	1,24E+00	3,37E+00	6,25E+00
	II	1,92E-01	8,36E-01	2,67E+00	7,03E+00
	III	1,42E-01	6,12E-01	1,98E+00	5,45E+00
5	I	4,75E-01	1,29E+00	3,11E+00	5,68E+00
	II	2,96E-01	8,73E-01	2,91E+00	7,26E+00
	III	2,14E-01	5,94E-01	2,27E+00	6,27E+00
6	I	8,24E-01	1,57E+00	2,95E+00	4,51E+00
	II	8,59E-01	1,37E+00	3,32E+00	6,95E+00
	III	5,32E-01	9,80E-01	2,56E+00	5,60E+00

7. CONCLUSION

From the simulation it can be seen that the method using a single redundant modulo reduces the area used, but the running time of the algorithm increases significantly. Using the approximate method improves the performance significantly, but has a larger area used.

Adapting the approximate method based on CRT to the single redundant modulo method can significantly improve the computational speed compared to the single redundant modulo method and reduce the area used compared to the approximate method with two redundant moduli. However, the disadvantage of this method is the limitation imposed by the redundant modulo.

One possible application of this approach is distributed data storage systems, such as the one described in Russian patent 2780148. In this case, in order to ensure reliability, the residue of the redundant modulo may be stored in the user's storage, while the residues of the working moduli are distributed to cloud storage. Thus, in case of an error or non-receipt of one of the file parts from the distributed environment, the original data can be restored in the manner described above.

A direction for further research may be to develop a single redundant modulo error correction method where the error on the control modulo can also be corrected.

ACKNOWLEDGEMENTS

This work was supported by the Russian Science Foundation 19-71-10033, https://rscf.ru/project/19-71-10033/.

REFERENCES

1. Tolstyakov, V. S. (1972) *Obnaruzhenie i ispravlenie oshibok v diskretnyh ustrojstvah* [Detection and correction of errors in discrete devices]. Moscow, USSR: Sov. radio, [in Russian].
2. Stakhov, A. P. (1977) *Vvedenie v algoritmicheskuyu teoriyu izmereniya* [Introduction to the algorithmic theory of measurement]. Moscow, USSR: Sov. radio, [in Russian].
3. Akushsky, I. Ya. & Yuditsky, D. I. (1968) *Mashinnaya arifmetika v ostatochnyh klassah* [Machine arithmetic in residual classes]. Moscow, USSR: Sov. radio, [in Russian].

4. Diamond, J. M. (1955). Checking codes for digital computers, *Proceedings of the Institute of Radio Engineers*, **43**(4), 487–488.
5. Brown, D. T. (1960). Error detecting and correcting binary codes for arithmetic operations, *IRE Transactions on Electronic Computers*, **3**, 333–337.
6. Chang, C. H., et al. (2015) Residue number systems: A new paradigm to datapath optimization for low-power and high-performance digital signal processing applications, *IEEE circuits and systems magazine*, **15**(4), 26–44.
7. Haron, N. Z. & Hamdioui, S. (2011) Redundant residue number system code for fault-tolerant hybrid memories, *ACM journal on emerging technologies in computing systems (JETC)*, **7**(1), 1–19.
8. Tay T. F. & Chang, C. H. (2017) Fault-tolerant computing in redundant residue number system, *Embedded systems design with special arithmetic and number systems*, 65–88.
9. Aremu, I. A. & Gbolagade, K. A. (2017) Redundant residue number system based multiple error detection and correction using Chinese remainder theorem (CRT), *Software Engineering*. 5(5), 72–80.
10. Gladkov, A., et al. (2022) Modified Error Detection and Localization in the Residue Number System, *Programming and Computer Software*, **48**(8), 598–605.
11. Soderstrand, M., Vernia, C. & Chang, J. H. (1983) An improved residue number system digital-to-analog converter, *IEEE transactions on circuits and systems*, **30**(12), 903–907.
12. Cherviakov, N. I. (2011). Metody, algoritmy i tekhnicheskaya realizaciya osnovnyh problemnyh operacij, vypolnyaemyh v sisteme ostatochnyh klassov [Methods, algorithms and technical implementation of the main problem operations performed in the residue number system], *Infokommunikacionnye tekhnologii*, **9**(4), 4–12. [in Russian]
13. Cherviakov, N. I., Babenko, M. G., Lyakhov, P. A. & Lavrinenko, I. N. (2013) Priblizhennyj metod opredeleniya znaka chisla v sisteme ostatochnyh klassov i ego tekhnicheskaya realizaciya [Approximate method for determining the sign of a number in the residue number system and its technical implementation]. *Nauchno-tekhnicheskie vedomosti Sankt-Peterburgskogo gosudarstvennogo politekhnicheskogo universiteta. Informatika. Telekommunikacii. Upravlenie*. **4**(176), 131–141. [in Russian]