Method for Computing Exponentiation Modulo the Positive and Negative Integers

Victor Krasnobayev[a], Alina Yanko[b], Anatolii Martynenko[b], Dmytro Kovalchuk[a]

[a]*V. N. Karazin Kharkiv National University, Svobody sq., 4, Kharkiv, 61022, Ukraine*
[b]*National University «Yuri Kondratyuk Poltava Polytechnic», Poltava, Ukraine*

Abstract

Numerous publications in recent years indicate that the prospects for creating high-speed computer systems (CS) based on the use of a modular number system (MNS) open up wide opportunities for using CSs with a high degree of parallelization of the processing of integer data. The application of the main properties of MNS and the possibility of using the tabular principle of data processing significantly increases the speed of performing integer arithmetic operations in comparison with the traditional binary positional number system. Research in this area shows the effectiveness of using MNS to increase the speed of execution besides to the arithmetic operations of addition, subtraction and multiplication of integers and the operation of exponentiating of the integers. However, until now there are no effective methods for exponentiating numbers in the MNS in all numeric domain (positive and negative). Therefore, this article developed a system of mathematical ratios that describe the researched process, based on which a method for for exponentiating numbers is developed, which unlike the known ones, can be implemented in the negative numeric domain. On the basis of the obtained method, algorithms for exponentiating numbers in the MNS were obtained, according to which devices for their implementation were synthesized.

Keywords

Arithmetic operation, computer system, system of mathematical ratios, method for exponentiating numbers, modular number system, modular structure, positional number system, tabular multiplication code.

1. Introduction

It should be noted that there is a numerous class of problems and algorithms where, in addition to performing integer basic arithmetic operations and the operation of exponentiating integers modulo in a positive numeric domain, there is a need to implement the above operations in a negative numeric domain. The absence of methods for exponentiating numbers represented in the MNS, both in positive and negative numerical areas, significantly narrows the area of effective use of the MNS as a number system of the CS [1-4].

The operation of exponentiating integers modulo has a wide practical application in cryptography and computer science. Below are a few examples:

- RSA encryption: RSA encryption uses modulo exponentiation to secure messages. In this system, each user has a public and a private key. When sending a message, the user encrypts it with the recipient's public key using the modulo exponentiation operation. The recipient, knowing his secret key, decrypts the message, also using the exponentiation modulo operation.

ICST-2023: Information Control Systems & Technologies, September 21-23, 2023, Odesa, Ukraine
EMAIL: v.a.krasnobaev@gmail.com (V. Krasnobayev); al9_yanko@ukr.net (A. Yanko); martynenko@pntu.edu.ua (A. Martynenko); kovalchuk.d.n@ukr.net (D. Kovalchuk)
ORCID: 0000-0001-5192-9918 (V. Krasnobayev); 0000-0003-2876-9316 (A. Yanko); 0000-0002-9576-0138 (A. Martynenko); 0000-0002-8229-836X (D. Kovalchuk)

- Diffie-Hellman algorithm: the operation of raising integers to a power is used to create a shared secret key between two participants. Participants choose a random number and the raising to a power of this number is performed using the public key of another participant, after which the resulting number is transmitted to another participant. Then other participant also raises the resulting number to the power of its private key to get the shared secret key.
- ElGamal algorithm: operation of raising to a power used to encrypt and decrypt messages. When encryption, the sender randomly chooses a number and raises the recipient's public key to the power of that number, and also raises the plaintext of the message to the power of the sender's private key. The results of the operations are multiplied to get the ciphertext. When decrypting, the recipient raises the first part of the ciphertext to the power of their private key, then uses that number to divide the second part of the ciphertext to get the original plaintext.
- Hash functions: Hash functions are widely used in computer science for data protection and integrity checking. One of the more popular hash functions, SHA-256, uses modulo exponentiation to compute a hash value.
- The operation of exponentiation modulo is also widely used in banking. For example, SSL uses RSA to encrypt data between a browser and a server.
- In addition, the operation of exponentiating integers modulocan be used in data networks to speed up calculations and reduce the amount of transmitted information.
- Pseudo-random number generation: Pseudo-random number generators often use exponentiation modulo to generate a sequence of random numbers. One of the most popular algorithms for generating pseudo-random numbers is the Mersenne algorithm, which uses the operation of exponentiation modulo [5-7].

In general, the operation of exponentiating integers modulo is one of the key elements of many cryptographic algorithms and has many practical applications in computer science and other fields. At the moment, many modern programming languages do not have tools (operators) that can implement the operation of exponentiating numbers, it is especially difficult to implement this operation for negative numbers [6].

So, research is aimed at developing a method for exponentiating numbers modulo MNS are relevant and important. However, practical methods cannot be used to perform the exponentiating operation in the negative numeric domain [8]. The article gives a method for performing this operation in various numeric domains.

By the type of the original number presented in the MNS $C_{MNS} = (c_1 \| c_2 \| ... \| c_{k-1} \| c_k \| c_{k+1} \| ... \| c_l)$, where $\|$ – mathematical sign of the concatenation operation: gluing operation, joining operation; it is difficult to attribute the number to any of the numeric domains. Consider the options of finding numbers in the MNS in the required numeric domains.

The first option. The original number represented in the MNS C_{MNS} has an additional two (or one) sign bits $\Psi_{+C_{MNS}}$ and $\Psi_{-C_{MNS}}$, where:

$$\Psi_{+C_{MNS}} = \begin{cases} 1, if\ C_{MNS} > 0, \\ 0, if\ C_{MNS} < 0; \end{cases}$$

$$\Psi_{-C_{MNS}} = \begin{cases} 0, if\ C_{MNS} > 0, \\ 1, if\ C_{MNS} < 0. \end{cases}$$

In this case, the original number in the MNS will be represented as $C_{MNS} = [\Psi_{+C_{MNS}} \| \Psi_{-C_{MNS}} \| (c_1 \| c_2 \| ... \| c_{k-1} \| c_k \| c_{k+1} \| ... \| c_l)]$. For this option, it is technically difficult to determine the sign of the result of the operation [3].

The second option. When performing the operation of exponentiating numbers by an arbitrary modulo MNS in all numeric domain, it is supposed to convert the number C_{MNS} in a modular structure (MS) $C_{MNS}^{\rightarrow}$ [1]:

$$\begin{cases} C_{MNS}^{\rightarrow} = \frac{1}{2}D + |C_{MNS}|, \text{ if } C_{MNS} \geq 0, \\ C_{MNS}^{\rightarrow} = \frac{1}{2}D - |C_{MNS}|, \text{ if } C_{MNS} < 0, \end{cases}$$

i.e. for positive numbers: $C_{MNS}^{\rightarrow} = \frac{1}{2}D + |C_{MNS}|$ and for negative: $C_{MNS}^{\rightarrow} = \frac{1}{2}D - |C_{MNS}|$, where $D = \prod_{k=1}^{l} p_k$, p_k – an arbitrary MNS module.

The main purpose of the article to develop an effective system of mathematical ratios (SMR) of the process of exponentiating numbers in the MNS in all numeric domain.

2. State of the art

Research conducted in recent decades in the field of development of an effective number system at the level of the arithmetic-logical device of the CS by a number of authors (Valah M., Svoboda A., Sabo N., Akushsky I. Ya., Yudytskyi D. I., Nikolaychuk Y. M., Dolgov O. I., Torgashov V. A ., Amerbaev V. M., etc.) proved that the application of the MNS as a CS number system showed that the use of the MNS as a CS number system to perform basic arithmetic operations (addition, subtraction and multiplication between integers and real numbers) significantly increases the speed of implementation of the above operations. Research in the field of the application of the non-positional number system (NPNS), to which the MNS belongs, shows that its practical use in the CS allows to significantly increase the productivity of the realization of arithmetic operations. It should be noted that there is a class of problems and algorithms where in addition to performing an integer arithmetic operations (addition, subtraction, multiplication), there is a need to implement the operation of exponentiating numbers in all numeric domain. This problem is discussed in the works of scientists Dr. Dimitrios Schinianakis and Thanos Stouraitis, who explored MNS in cryptography. In particular, in work [9] providing a detailed explanation of cryptographic algorithms based on the MNS, including RSA, ElGamal, and elliptic curve cryptography.

They discuss the problem and limitation of using the MNS in cryptography, such as the difficulty of handling negative numbers. Various methods have been explored to solve this problem and ensure the reliability of cryptographic systems based on the system of residual classes.

Thus, the unsolvedness of the above-listed problem, connected primarily with the operation of exponentiating numbers in the negative numeric domain in MNS, determined the purpose of the article.

The principle of realization of the system of mathematical ratios is developed using the main internals of the NPNS that define the non-positional code data structure of the MNS which ensures high productivity (speed) and reliability of arithmetic operations for the implementation of computational algorithms in the CS, consisting of a set of arithmetic (modular) operations [10]. A distinctive feature of this article is that the proposed system of mathematical ratios of the process of exponentiating numbers in all numeric domain will significantly expand the scope of MNS and increase the speed of this operation in CS, and the method are reduced to algorithms, on the basis of which classes of patent-eligible devices for which Ukrainian patents have been obtained have been developed [11, 12].

3. Realization of computing exponentiation of numbers in the MNS in all numeric domain

A system of mathematical ratios the process of exponentiating numbers was developed based on an analytical ratio $(C_{MNS}^{r})^{\rightarrow} = f(C_{MNS}^{\rightarrow})$ which defines the relation of the result C_{MNS}^{r} number exponentiating operations C_{MNS} in the MNS (to the degree of *r*) presented in the MS, from the number $C_{MNS}^{\rightarrow}$ immediately in the MS [13].

Let's define SMR $(C_{MNS}^{r})^{\rightarrow} = f(C_{MNS}^{\rightarrow})$ for values $r = 2$ and $p_1 = 2$. In this case, in the MNS we have that:

$$D = \prod_{k=1}^{l} p_k = (0 \| 0 \| ... \| 0 \| ... \| 0), \tag{1}$$

$$\frac{1}{2} D = \prod_{k=2}^{l} p_k = (1 \| 0 \| ... \| 0 \| ... \| 0), \tag{2}$$

where l – the number of bases of the MNS.

According to the representation of the MS numbers in the MNS, we have that:

$$\begin{cases} C_{MNS}^{\rightarrow} = \frac{1}{2} D + C_{MNS}, \\ (C_{MNS}^{r})^{\rightarrow} = \frac{1}{2} D + C_{MNS}^{r}. \end{cases} \tag{3}$$

When changing the domain of numbers C_{MNS} and $C_{MNS}^{\rightarrow}$ from the mathematical expression (3) can be defined as:

$$(C_{MNS}^{r})^{\rightarrow} = \left(\frac{1}{2} D + C_{MNS}^{r} \right) \bmod D. \tag{4}$$

Let's carry out the following numerical transformations:

$$(C_{MNS}^{\rightarrow})^2 = C_{MNS}^{\rightarrow} \cdot C_{MNS}^{\rightarrow} = (\frac{1}{2} D + C_{MNS}) \cdot (\frac{1}{2} D + C_{MNS}) = C_{MNS}^2 + C_{MNS} \cdot D + \frac{1}{2} D \cdot \frac{1}{2} D. \tag{5}$$

In this case, expression (5) will be presented as:

$$(C_{MNS}^{\rightarrow})^2 = C_{MNS}^2 + \frac{1}{2} D. \tag{6}$$

On the other hand, we have that:

$$\begin{cases} (C_{MNS}^{\rightarrow})^2 = \frac{1}{2} D + C_{MNS}^2, \\ C_{MNS}^2 = (C_{MNS}^{\rightarrow})^2 - \frac{1}{2} D. \end{cases} \tag{7}$$

By replacing the value of C_{MNS}^2 from relation (7) into relation (6) can obtain that:

$$(C_{MNS}^{\rightarrow})^2 = (C_{MNS}^2)^{\rightarrow} - \frac{1}{2} D + \frac{1}{2} D \quad \Rightarrow \quad (C_{MNS}^{\rightarrow})^2 = (C_{MNS}^2)^{\rightarrow}. \tag{8}$$

Mathematical ratio (8) is the SMR of squaring numbers modulo MNS. It is also possible to get SMR for the general case, when $r > 2$ [14] in the form $(C_{MNS}^{r})^{\rightarrow} = (C_{MNS}^{\rightarrow})^{r}$.

In this case, it is obvious that:

$$(C_{MNS}^{r})^{\rightarrow} = (C_{MNS}^{r-1})^{\rightarrow} \cdot C_{MNS}^{\rightarrow}, \tag{9}$$

Mathematical ratio (9) is a generalized SMR process of exponentiating numbers modulo MNS.

For the convenience of applying ratio (9), it is sometimes possible to use the ratio:

$$\begin{aligned} &[c_1^{r} (\bmod p_1) \| c_2^{r} (\bmod p_2) \| ... \| c_k^{r} (\bmod p_k) \| ... \| c_l^{r} (\bmod p_1)]^{\rightarrow} = \\ &= (c_1^{\rightarrow} \| c_2^{\rightarrow} \| ... \| c_{k-1}^{\rightarrow} \| c_k^{\rightarrow} \| c_{k+1}^{\rightarrow} \| ... \| c_l^{\rightarrow})^{r} \end{aligned} \tag{10}$$

Processed numbers C_{MNS}^{r} and $(C_{MNS}^{\rightarrow})^{r}$ are in the range:

$$\begin{cases} -\frac{1}{2} D \le C_{MNS}^{r} \le \frac{1}{2} (D - 1), \\ 0 \le (C_{MNS}^{\rightarrow})^{r} \le D - 1. \end{cases}$$

Based on SMR (9) implementation of the operation of exponentiating numbers, consider the method of tabular (matrix) realization of the operation of exponentiating numbers modulo MNS in all numeric domain [15-18]. With a tabular realization of the operation of exponentiating numbers modulo p_k MNS, the residues of the number $C_{MNS}^{\rightarrow}$ are encoded by the tabular multiplication code

(TMC) [13] as follows $c_k^{\rightarrow} = [\mu_{c_k}^{\rightarrow} \| (c_k^{\rightarrow})^*]$. The sign $\mu_{c_k}^{\rightarrow}$ of the TMC can be represented in the following form.

For p_k – an even number:

$$\mu_{c_k}^{\rightarrow} = \begin{cases} 0, if\ 0 \le c_k^{\rightarrow} \le \frac{p_k}{2}, \\ 1, if\ \frac{p_k}{2} < c_k^{\rightarrow} \le p_k - 1. \end{cases} \quad (11)$$

For p_k – an odd number:

$$\mu_{c_k}^{\rightarrow} = \begin{cases} 0, if\ 0 \le c_k^{\rightarrow} \le \frac{(p_k - 1)}{2}, \\ 1, if\ \frac{(p_k - 1)}{2} < c_k^{\rightarrow} \le p_k - 1. \end{cases} \quad (12)$$

The numerical part of $(c_k^{\rightarrow})^*$ TMC of the residue $c_k^{\rightarrow}$ is determined as follows.

For p_k – an even number:

$$(c_k^{\rightarrow})^* = \begin{cases} c_k^{\rightarrow}, if\ 0 \le c_k^{\rightarrow} \le \frac{p_k}{2}; \\ \overline{c_k^{\rightarrow}} = p_k - c_k^{\rightarrow}, if\ \frac{p_k}{2} < c_k^{\rightarrow} \le p_k - 1, \end{cases} \quad (13)$$

wherein $0 \le (c_k^{\rightarrow})^* \le \frac{p_k}{2}$.

For p_k – an odd number:

$$(c_k^{\rightarrow})^* = \begin{cases} c_k^{\rightarrow}, if\ 0 \le c_k^{\rightarrow} \le \frac{(p_k - 1)}{2}; \\ \overline{c_k^{\rightarrow}} = p_k - c_k^{\rightarrow}, if\ \frac{(p_k - 1)}{2} < c_k^{\rightarrow} \le p_k - 1, \end{cases} \quad (14)$$

wherein $0 \le (c_k^{\rightarrow})^* \le \frac{(p_k - 1)}{2}$.

Result $(c_k^{\rightarrow} \cdot c_k^{\rightarrow}) \bmod p_k$ residue multiplication operations $c_k^{\rightarrow}$ on itself modulo p_k submitted to TMC, i.e. in the form $\left\{\mu_k^{\rightarrow} \| [(c_k^{\rightarrow})^* (c_k^{\rightarrow})^*] \bmod p_k\right\}$. Then the condition performed $(\mu_{c_k}^{\rightarrow} + \mu_{c_k}^{\rightarrow}) = 0 \pmod 2$. In this case, we have that:

$$(c_k^{\rightarrow} \cdot c_k^{\rightarrow}) \bmod p_k = [(c_k^{\rightarrow})^* \cdot (c_k^{\rightarrow})^*] \bmod p_k, \quad (15)$$

wherein $0 \le [(c_k^{\rightarrow})^* \cdot (c_k^{\rightarrow})^*] \bmod p_k \le p_k - 1$.

Based on the developed SMR (8) and using the tabular realization of the modular multiplication operation [13, 19-22], the article improves the method of exponentiating numbers modulo MNS in all numeric domain.

The developed method. A method for exponentiating numbers in the MNS in all numeric domain consists of the following stages:

1. Set the initial numbers for the realization of the method for exponentiating number $C_{MNS} = (c_1 \| c_2 \| ... \| c_{k-1} \| c_k \| c_{k+1} \| ... \| c_l)$, an arbitrary modulo $p_k\ (k = \overline{1, l})$ MNS (to the degree of *r*).

2. Coding of initial numbers C_{MNS} into code words [23] presented in the MS of the form $C_{MNS}^{\rightarrow}$:

$$\begin{cases} C_{MNS}^{\rightarrow} = \frac{1}{2} D + |C_{MNS}|, if\ C \ge 0, \\ C_{MNS}^{\rightarrow} = \frac{1}{2} D - |C_{MNS}|, if\ C < 0, \end{cases}$$

$$\begin{cases} -\frac{1}{2}D \le C_{MNS} \le \frac{1}{2}(D-1), \\ 0 \le C_{MNS}^{\rightarrow} \le D-1. \end{cases} \quad \begin{cases} (C_{MNS}^{r})^{\rightarrow} = \frac{1}{2}D + \left|C_{MNS}^{r}\right|, if\ C_{MNS}^{r} \ge 0, \\ (C_{MNS}^{r})^{\rightarrow} = \frac{1}{2}D - \left|C_{MNS}^{r}\right|, if\ C_{MNS}^{r} < 0, \end{cases} \quad \begin{cases} -\frac{1}{2}D \le C_{MNS}^{r} \le \frac{1}{2}(D-1), \\ 0 \le (C_{MNS}^{\rightarrow})^{r} \le D-1. \end{cases}$$

3. Representation of the residues $c_k^{\rightarrow}$ of the number $C_{MNS}^{\rightarrow} = (c_1^{\rightarrow} \| c_2^{\rightarrow} \| ... \| c_{k-1}^{\rightarrow} \| c_k^{\rightarrow} \| c_{k+1}^{\rightarrow} \| ... \| c_l^{\rightarrow})$ in the MS by modules $p_k\ (k = \overline{1,l})$ based on the apply of TMC $c_k^{\rightarrow} = [\mu_{c_k}^{\rightarrow} \| (c_k^{\rightarrow})^{*}]$, where:

for p_k – an even number: $\mu_{c_k}^{\rightarrow} = \begin{cases} 0, if\ 0 \le c_k^{\rightarrow} \le \frac{p_k}{2}, \\ 1, if\ \frac{p_k}{2} < c_k^{\rightarrow} \le p_k - 1, \end{cases}$

$$(c_k^{\rightarrow})^{*} = \begin{cases} c_k^{\rightarrow}, if\ 0 \le c_k^{\rightarrow} \le \frac{p_k}{2}; \\ \overline{c_k^{\rightarrow}} = p_k - c_k^{\rightarrow}, if\ \frac{p_k}{2} < c_k^{\rightarrow} \le p_k - 1, \end{cases}$$

wherein $0 \le (c_k^{\rightarrow})^{*} \le \frac{p_k}{2}$;

for p_k – an odd number: $\mu_{c_k}^{\rightarrow} = \begin{cases} 0, if\ 0 \le c_k^{\rightarrow} \le \frac{(p_k-1)}{2}, \\ 1, if\ \frac{(p_k-1)}{2} < c_k^{\rightarrow} \le p_k - 1, \end{cases}$

$$(c_k^{\rightarrow})^{*} = \begin{cases} c_k^{\rightarrow}, if\ 0 \le c_k^{\rightarrow} \le \frac{(p_k-1)}{2}; \\ \overline{c_k^{\rightarrow}} = p_k - c_k^{\rightarrow}, if\ \frac{(p_k-1)}{2} < c_k^{\rightarrow} \le p_k - 1, \end{cases}$$

wherein $0 \le (c_k^{\rightarrow})^{*} \le \frac{(p_k-1)}{2}$;

4. Definition of result $(c_k^{\rightarrow})^2 = (c_k^{\rightarrow} \cdot c_k^{\rightarrow}) \bmod p_k$, $(k = \overline{1,l})$ modular multiplication operations in the form $\mu_k^{\rightarrow} \| [(c_k^{\rightarrow}) \cdot (c_k^{\rightarrow})] \bmod p_k$.

5. Determining the result of an operation $\left[(C_{MNS}^{\rightarrow})^{r-1} \cdot C_{MNS}^{\rightarrow}\right] \bmod D = \left\{\left[(c_1^{\rightarrow})^{r-1}\right] \bmod p_1 \| \right.$ $\| \left[(c_2^{\rightarrow})^{r-1}\right] \bmod p_2 \| ... \| \left[(c_2^{\rightarrow})^{r-1}\right] \bmod p_k \| ... \| \left.\left[(c_l^{\rightarrow})^{r-1}\right] \bmod p_l \right\} \cdot (c_1^{\rightarrow} \| c_2^{\rightarrow} \| ... \| c_{k-1}^{\rightarrow} \| c_k^{\rightarrow} \| c_{k+1}^{\rightarrow} \| ... \| c_l^{\rightarrow})$ exponentiating numbers c_k of integer $C_{MNS} = (c_1 \| c_2 \| ... \| c_{k-1} \| c_k \| c_{k+1} \| ... \| c_l)$ modulo $p_k\ (k = \overline{1,l})$ MNS to the degree *r*.

6. According to the SMR: $\left[c_1^r (\bmod p_1) \| ... \| c_k^r (\bmod p_k) \| ... \| c_l^r (\bmod p_l)\right]^{\rightarrow} = (c_1^{\rightarrow} \| c_2^{\rightarrow} \| ...$ $... \| c_{k-1}^{\rightarrow} \| c_k^{\rightarrow} \| c_{k+1}^{\rightarrow} \| ... \| c_l^{\rightarrow})^r$ the process of exponentiating numbers modulo, the operation is implemented:

$$\left[(C_{MNS}^{\rightarrow})^{r-1} \cdot C_{MNS}^{\rightarrow}\right] \bmod D = \left\{\left[(c_1^{\rightarrow})^{r-1}\right] \bmod p_1 \| \left[(c_2^{\rightarrow})^{r-1}\right] \bmod p_2 \| ... \right.$$

$$... \| \left[(c_k^{\rightarrow})^{r-1}\right] \bmod p_k \| ... \| \left.\left[(c_l^{\rightarrow})^{r-1}\right] \bmod m_l \right\} \cdot (c_1^{\rightarrow} \| c_2^{\rightarrow} \| ... \| c_{k-1}^{\rightarrow} \| c_k^{\rightarrow} \| c_{k+1}^{\rightarrow} \| ... \| c_l^{\rightarrow})$$

exponentiating numbers modulo p_k MNS in all numeric domain.

4. Results

The result of the research is induced in the form of using the developed method for exponentiating numbers for the MNS with bases $p_1 = 2,\ p_2 = 5,\ p_3 = 7,$ wherein $D = 70.$ Total volume of positive codewords C_{MNS} in the MNS presented in Table 1, where C_{PNS} – number in the positional number system (PNS). Table 2 submitts the original positive numbers C_{MNS} and numbers in MS $C_{MNS}^{\rightarrow}$.

Table 1

The numbers C_{MNS} in the MNS

C_{PNS}	C_{MNS}			C_{PNS}	C_{MNS}		
	$p_1 = 2$	$p_2 = 5$	$p_3 = 7$		$p_1 = 2$	$p_2 = 5$	$p_3 = 7$
0	0	0	0	35	1	0	0
1	1	1	1	36	0	1	1
2	0	2	2	37	1	2	2
3	1	3	3	38	0	3	3
4	0	4	4	39	1	4	4
5	1	0	5	40	0	0	5
6	0	1	6	41	1	1	6
7	1	2	0	42	0	2	0
8	0	3	1	43	1	3	1
9	1	4	2	44	0	4	2
10	0	0	3	45	1	0	3
11	1	1	4	46	0	1	4
12	0	2	5	47	1	2	5
13	1	3	6	48	0	3	6
14	0	4	0	49	1	4	0
15	1	0	1	50	0	0	1
16	0	1	2	51	1	1	2
17	1	2	3	52	0	2	3
18	0	3	4	53	1	3	4
19	1	4	5	54	0	4	5
20	0	0	6	55	1	0	6
21	1	1	0	56	0	1	0
22	0	2	1	57	1	2	1
23	1	3	2	58	0	3	2
24	0	4	3	59	1	4	3
25	1	0	4	60	0	0	4
26	0	1	5	61	1	1	5
27	1	2	6	62	0	2	6
28	0	3	0	63	1	3	0
29	1	4	1	64	0	4	1
30	0	0	2	65	1	0	2
31	1	1	3	66	0	1	3
32	0	2	4	67	1	2	4
33	1	3	5	68	0	3	5
34	0	4	6	69	1	4	6

Table 2

The numbers $C^{\rightarrow}$ in the MS

C	$C^{\rightarrow}$	C	$C^{\rightarrow}$	C	$C^{\rightarrow}$	C	$C^{\rightarrow}$	C	$C^{\rightarrow}$
-35	0	-21	14	-7	28	7	42	21	56
-34	1	-20	15	-6	29	8	43	22	57
-33	2	-19	16	-5	30	9	44	23	58
-32	3	-18	17	-4	31	10	45	24	59
-31	4	-17	18	-3	32	11	46	25	60
-30	5	-16	19	-2	33	12	47	26	61
-29	6	-15	20	-1	34	13	48	27	62
-28	7	-14	21	0	35	14	49	28	63
-27	8	-13	22	1	36	15	50	29	64
-26	9	-12	23	2	37	16	51	30	65
-25	10	-11	24	3	38	17	52	31	66
-24	11	-10	25	4	39	18	53	32	67
-23	12	-9	26	5	40	19	54	33	68
-22	13	-8	27	6	41	20	55	34	69

The practical use of the developed method for the MNS with bases $p_1 = 2,\ p_2 = 5,\ p_3 = 7,$ $D = \prod_{k=1}^{l} p_k = p_1 \cdot p_2 \cdot p_3 = 2 \cdot 5 \cdot 7 = 70$ is presented in the form of examples.

Example 1. Let $C_{MNS} = 2 = (0 \| 2 \| 2)$ and $r = 2$. Let's define the value of $C_{MNS}^{r} = 2^2$. Because $C_{MNS} = 2 > 0,$ then we get that $C_{MNS}^{\rightarrow} = \frac{1}{2}D + C_{MNS} = 35 + 2 = 37 = (1 \| 0 \| 0) + (0 \| 2 \| 2) =$ $= (1 \| 2 \| 2) = 37.$ Because $r = 2$ as a result of multiplying the value of $C_{MNS}^{\rightarrow} = (1 \| 2 \| 2)$ by itself $1 \cdot 1 = 1 (\mathrm{mod}\, 2),\ 2 \cdot 2 = 4 (\mathrm{mod}\, 5)$ and $2 \cdot 2 = 4 (\mathrm{mod}\, 7),$ we get that $(C_{MNS}^{\rightarrow})^2 = (1 \| 4 \| 4) = 39.$

Check: $(C_{MNS}^{\rightarrow})^2 = 37^2 = 37 \times 37 = 1369 = 39 (\mathrm{mod}\, 70) = (1 \| 2 \| 2) \times (1 \| 2 \| 2) = (1 \| 4 \| 4) = 39.$

$(C_{MNS}^{2})^{\rightarrow} = \frac{1}{2}D + C_{MNS}^{2},\ C_{MNS}^{2} = (C_{MNS}^{2})^{\rightarrow} - \frac{1}{2}D,\ 2^2 = 39 - 35 = 4,\ 2^2 = 4.$

Example 2. Let $C_{MNS} = -2\ \left[2 = (0 \| 2 \| 2)\right]$ and $r = 2$. Because $C_{MNS} = -2 < 0,$ then $C_{MNS}^{\rightarrow} = \frac{1}{2}D - C_{MNS} = 35 - 2 = 33 = (1 \| 0 \| 0) - (0 \| 2 \| 2) = (1 \| 3 \| 5) = 33.$ Because $r = 2$ we have $1 \cdot 1 = 1 (\mathrm{mod}\, 2),\ 3 \cdot 3 = 4 (\mathrm{mod}\, 5),\ 5 \cdot 5 = 4 (\mathrm{mod}\, 7),$ we get that $(C_{MNS}^{\rightarrow})^2 = (1 \| 4 \| 4) = 39.$

Check: $(C_{MNS}^{\rightarrow})^2 = 33^2 = 33 \times 33 = 1089 = 39 (\mathrm{mod}\, 70) = (1 \| 3 \| 5) \times (1 \| 3 \| 5) = (1 \| 4 \| 4) = 39.$

$C_{MNS}^{2} = (C_{MNS}^{2})^{\rightarrow} - \frac{1}{2}D,\ (-2)^2 = 39 - 35 = 4,\ (-2)^2 = 4.$

Example 3. Let $C_{MNS} = -2\ \left[2 = (0 \| 2 \| 2)\right]$ and $r = 3$. Because $C_{MNS} = -2 < 0,$ then we have that $C_{MNS}^{\rightarrow} = \frac{1}{2}D - C_{MNS} = 35 - 2 = 33 = (1 \| 0 \| 0) - (0 \| 2 \| 2) = (1 \| 3 \| 5) = 33.$ The first iteration of the multiplication gives the result: $C_{MNS}^{\rightarrow} \times C_{MNS}^{\rightarrow} = (C_{MNS}^{\rightarrow})^2,\ 1 \cdot 1 = 1 (\mathrm{mod}\, 2),\ 3 \cdot 3 = 4 (\mathrm{mod}\, 5),$ $5 \cdot 5 = 4 (\mathrm{mod}\, 7).$ The second iteration (because $r = 3$): $(C_{MNS}^{\rightarrow})^2 \times C_{MNS}^{\rightarrow}.$ In this case we have $(C_{MNS}^{\rightarrow})^2 = (1 \| 4 \| 4),\ (C_{MNS}^{\rightarrow})^3 = (C_{MNS}^{\rightarrow})^2 \times C_{MNS}^{\rightarrow} = (1 \| 4 \| 4) \times (1 \| 3 \| 5) = (1 \| 2 \| 6) = 27.$

Check: $(C_{MNS}^{\rightarrow})^3 = 33^3 = 35937 = 27 (\mathrm{mod}\, 70) = C_{MNS}^{\rightarrow} \times C_{MNS}^{\rightarrow} \times C_{MNS}^{\rightarrow} = (1 \| 3 \| 5) \times (1 \| 3 \| 5) \times (1 \| 3 \| 5) =$ $= (1 \| 2 \| 6) = 27.\ C_{MNS}^{3} = (C_{MNS}^{3})^{\rightarrow} - \frac{1}{2}D,\ (-2)^3 = 27 - 35,\ (-2^3) = -8.$

Example 4. Let $C_{MNS} = -3\ [3 = (1\|3\|3)]$ and $r = 3$. Because $C_{MNS} = -3 < 0$, then $C^{\rightarrow}_{MNS} = \frac{1}{2}D - C_{MNS} = 35 - 3 = 32 = (1\|0\|0) - (1\|3\|3) = (0\|2\|4) = 32$. The first iteration of the multiplication gives the result: $C^{\rightarrow}_{MNS} \times C^{\rightarrow}_{MNS} = (C^{\rightarrow}_{MNS})^2 = (0\|2\|4) \times (0\|2\|4) = (0\|4\|2)$. The second iteration (because $r = 3$): $(C^{\rightarrow}_{MNS})^2 \times C^{\rightarrow}_{MNS} = (C^{\rightarrow}_{MNS})^3 = (0\|4\|2) \times (0\|2\|4) = (0\|3\|1)$. Thus, we get that $C^{r}_{MNS} = (-3)^3 = (0\|3\|1) = 8$.

Check: $(C^{\rightarrow}_{MNS})^3 = 32^3 = 32768 = 8(\mathrm{mod}\,70) = C^{\rightarrow}_{MNS} \times C^{\rightarrow}_{MNS} \times C^{\rightarrow}_{MNS} = (0\|2\|4) \times (0\|2\|4) \times$ $\times (0\|2\|4) = (0\|3\|1) = 8$. $C^{3}_{MNS} = (C^{3}_{MNS})^{\rightarrow} - \frac{1}{2}D$, $(-3)^3 = 8 - 35 = -27$, $(-3)^3 = -27$.

5. Conclusion

The procedure for realization the operation of exponentiating numbers in the MNS in a positive numeric domain has been researched. As the operation of exponentiating numbers is one of the key operations of many algorithms and protocols that are used in modern CS (programming, cryptography, optimization algorithms, etc.), and there is no effective implementation of this operation, especially for negative numbers. Two options for representing numbers in the MNS are considered, in all numeric domain. The first option is as follows: the original number in the MNS has an additional two sign bits $\Psi_{+C_{MNS}}$ and $\Psi_{-C_{MNS}}$. These bits symbolize the sign of the number in the MNS. The second option is as follows: for perform the process of realizing the operation of exponentiating numbers by modulo MNS in all numeric domain, it is supposed to represent the original number in a modular structure. The article developed a method for exponentiating numbers in the MNS, both in positive and negative numeric domain. This method is based on the use of a synthesized mathematical model in an analytical ratio, which is a generalized system of mathematical ratios of the process of exponentiating numbers modulo MNS. The development of the method was carried out by applying a special coding of numbers in the MS using the tabular principle of data processing. The result of the developed method is presented in the form of examples of the operation of exponentiating numbers represented in the MNS. An analysis of the solution of examples showed the practical value of the developed method.

6. References

[1] I. Ya. Akushskii, D. I. Yuditskii, Machine Arithmetic in Residual Classes, Sov. Radio, Moscow, 1968.

[2] Y.M. Nikolaychuk (Ed.), Specialized computer technologies in informatics, TernogrMS, Ternopil, 2017.

[3] A. I. Kornilov, M. Yu. Semenov, V. S. Kalashnikov, Methods for Hardware Optimization of Adders for Two Operands in the System of Residual Classes, 1st. ed., Electronics, Moscow, 2017, pp. 75–82.

[4] A. Sasaki, The Basis for Implementation of Adidive Operations in the Residue Number System, IEEE Transactions on Computers C-17, (1968) 1066–1073. doi:10.1109/TC.1968.226466.

[5] Exponentiation by squaring, 2001. URL: https://en.wikipedia.org/wiki/Exponentiation_by_squaring

[6] A. Cui, H. Zhao, X. Zhang, B. Zhao, Z. Li, Power system real time data encryption system based on DES algorithm, in: 2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), IEEE Press, Beihai, 2021, pp. 220–228. doi: 10.1109/ICMTMA52658.2021.00056.

[7] V. Mainanwal, M. Gupta, S. K .Upadhayay, Zero Knowledge Protocol with RSA Cryptography Algorithm for Authentication in Web Browser Login System (Z-RSA), in: 2015 Fifth International Conference on Communication Systems and Network Technologies, IEEE Press, Gwalior, 2015, pp. 776–780. doi:10.1109/CSNT.2015.90.

[8] Ch. Pakkiraiah, R. Satyanarayana, Design and FPGA Realization of an Energy Efficient Artificial Neural Modular Exponentiation Architecture. Computing, Communication and Learning, (2023) 115–126. doi:10.1007/978-3-031-21750-0_10.
[9] D. Schinianakis, T. Stouraitis, Number Systems in Cryptography: Design, Challenges, Robustness, 2016. doi:10.1007/978-3-319-14971-4_4
[10] Weihang Tan, Sin-Wei Chiu, Antian Wang, Yingjie Lao, Keshab K. Parhi, PaReNTT: Low-Latency Parallel Residue Number System and NTT-Based Long Polynomial Modular Multiplication for Homomorphic Encryption, 2023. doi:10.48550/arXiv.2303.02237.
[11] V. Krasnobayev, S. Koshman, A. Yanko, A device for raising numbers represented in the residual class to the power of a natural number, 2014. Patent for useful model No. 95060, Filed June 18th., 2014, Issued December 10th., 2014.
[12] V. Krasnobayev, A. Zamula, S. Rassomakhin, A. Yanko, A device for squaring numbers in the residual class system, 2014. Patent for useful model No. 129249, Filed April 23th., 2014, Issued October 25th., 2014.
[13] V. Krasnobaev, A. Kuznetsov, V. Babenko, M. Denysenko, M. Zub, V. Hryhorenko, The Method of Raising Numbers, Represented in the System of Residual Classes to an Arbitrary Power of a Natural Number, in: 2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON), IEEE Press, Lviv, 2019, pp. 1133–1138. doi:10.1109/UKRCON.2019.8879793
[14] V. Krasnobayev, A. Kuznetsov, I. Lokotkova, A. Kiian, T. Kuznetsova, Techniques for Raising the Remainder to a Power in the System of Residual Classes, in: 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), IEEE Press, Kyiv, 2020, pp. 145–150. doi:10.1109/DESSERT50317.2020.9125049.
[15] Z. Xiaoqiang, Z. Xueheng, Image encryption algorithm based on the Matryoshka transform and modular-inverse matrix, 2023. doi:10.21203/rs.3.rs-2663096/v1.
[16] S. Banerjee, S. Chakraborty, N. Dey, A. Kumar Pal, R. Ray, High Payload Watermarking using Residue Number System. International Journal of Image, Graphics and Signal Processing 7 (2015) 1–8. doi: 10.5815/ijigsp.2015.03.01
[17] K.A. Gbolagade, S.D. Cotofana, Residue Number System operands to decimal conversion for 3-moduli sets, in: 2008 51st Midwest Symposium on Circuits and Systems, IEEE, 2008, pp. 222–224. doi:10.1109/MWSCAS.2008.4616918
[18] B. Parhami, On equivalences and fair comparisons among residue number systems with special moduli, in: 2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers, IEEE, 2010, pp. 1690–1694. doi:10.1109/ACSSC.2010.5757827.
[19] K. Phalakarn, A. Surarerks, Alternative Redundant Residue Number System Construction with Redundant Residue Representations, in: 2018 3rd International Conference on Computer and Communication Systems (ICCCS), IEEE, 2018, pp. 457–461. doi:10.1109/CCOMS.2018.8463305
[20] D. Younes, P. Steffan, Efficient image processing application using residue number system, in: Proceedings of the 20th International Conference Mixed Design of Integrated Circuits and Systems - MIXDES 2013, IEEE, Gdynia, 2013, pp. 468–472.
[21] V. Yatskiv, T. Tsavolyk, N. Yatskiv, The correcting codes formation method based on the residue number system, in: 2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), IEEE, 2017, pp. 237–240. doi:10.1109/CADSM.2017.7916124
[22] D.I. Popov, A.V. Gapochkin, Development of Algorithm for Control and Correction of Errors of Digital Signals, Represented in System of Residual Classes, in: 2018 International Russian Automation Conference (RusAutoCon), IEEE, Sochi, 2018, pp. 1–3. doi:10.1109/RUSAUTOCON.2018.8501826.
[23] H. Bombín, C. Dawson, Ye-Hua Liu, N. Nickerson, F. Pastawski, S. Roberts, Modular decoding: parallelizable real-time decoding for quantum computers, 2023.