Brousentsov's Ternary Principle, Bergman's Number System and Ternary Mirror-symmetrical Arithmetic

ALEXEY STAKHOV

P.O. Box 2878, Vinnitsa 27, Ukraine 21027
Email: anna@nest.vinnica.ua

We consider an original ternary number system called the ternary mirror-symmetrical number system in the article. It is a synthesis of the classical ternary symmetrical number system and the number system with an irrational base called Bergman's number system. The main engineering result is a development of an original matrix and pipeline ternary mirror-symmetrical adder, which can be used for fast 'pipeline' addition and multiplication in digital signal processors.

Received 19 December 2000; revised 15 May 2001

1. BROUSENTSOV'S TERNARY PRINCIPLE AND BERGMAN'S NUMBER SYSTEM

1.1. Ternary symmetrical number system

It is well known that the design of computers begins with the choice of number system, which determines many technical characteristics of computers. At the dawn of the computer era the choice of the optimal number system for electronic computers was solved brilliantly by the outstanding American physicist and mathematician John von Neumann, which gave emphatic preference to the binary number system in electronic computers. The famous John von Neumann Principles include three basic ideas of electronic computer design: binary number system, binary (Boolean) logic and binary memory element ('flip-flop').

However, research in the field of number systems has continued in modern computer science. A basic motivation of these investigations is connected with overcoming a number of essential deficiencies of the classical binary number system. The most well known of these are (a) the 'sign problem' (it is impossible to represent negative numbers and perform arithmetical operations over them in 'direct' code) that complicates arithmetical computer structures, and (b) the problem of zero redundancy (all binary code combinations are 'permitted') that does not allow the checking of informational processes in processors and computers effectively.

The first attempt to overcome the first of these deficiencies of the binary number system was made in Soviet science at the dawn of the computer era. The original computer project (the ternary 'Setun' computer) [1, 2, 3, 4, 5] designed in 1958 in Moscow University (the principal designer was Nikolay Brousentsov) was a clever example of an optimal solution of the sign problem. The so-called 'Brousentsov's Ternary Principle' of computer design was realized in the Setun computer, which is based on the ideas of ternary logic, ternary symmetrical number system and ternary memory element ('flip-flap-flop').

The 'ternary symmetrical number system' is a key idea of Brousentsov's Ternary Principle [1, 2, 3, 4, 5]. This uses the positional method of number representation

$$N = \sum_{i=1}^{n} b_i 3^{i-1} \tag{1}$$

where $b_i (i = 1, 2, \dots, n)$ is ternary numeral $\{\bar{1} = -1, 0$ and $1\}$ of the ith digit, 3^{i-1} is the 'weight' of the ith digit and the number 3 is the base of the number system.

The basic advantage of the number system (1) in comparison to the classical binary one with the numerals 0 and 1 is an elegant solution to the sign problem. The sign of a number is determined with the highest significant numeral of the ternary representation. For example, the number $N_1 = 0\bar{1}110\bar{1}$ is negative but the number $N_2 = 1\bar{1}0\bar{1}001$ is positive. Both positive and negative numbers are represented in 'direct' code and all arithmetical operations are realized in direct code. It is easy to get the representation of the negative number $(-N)$ from the ternary representation of positive number N using the rule of 'ternary inversion'

$$1 \to \bar{1}, \quad 0 \to 0, \quad \bar{1} \to 1. \tag{2}$$

Ternary-symmetrical addition and multiplication are based on the trivial identities connecting powers of number 3: $3^i + 3^i = 3^{i+1} - 3^i$ (for addition) and $3^m \times 3^n = 3^{m+n}$ (for multiplication) [6]. Ternary-symmetrical subtraction of numbers $A - B$ is reduced to addition of numbers $A +$

$(-B)$ if the rule of ternary inversion (2) is applied to the subtrahend B. The ternary-symmetrical division is similar to classical binary division and is reduced to divisor shift and its subtraction from the dividend [6].

The Setun computer had been designed on magnetic elements and therefore it did not get wide practical use but its architecture based on the ternary principle was so perfect that at the present time the project of the Setun computer still attracts the attention of many computer specialists. The famous Soviet computer specialist Professor Pospelov wrote in his book [6]:

> The barriers, which stand in the way of the ternary symmetrical number systems application to computers, are barriers of a technical character. Up to now economical and effective elements with three stable states have not been elaborated. As soon as such elements are developed a majority of computers of a universal kind and many special computers most probably will be designed so that they would function in the ternary symmetrical number system.

The famous American scientist Donald Knuth [7] also expressed the opinion that the replacement of 'flip-flop' by 'flip-flap-flop' would happen some day.

1.2. Tau System

Also at the dawn of the computer era, another original discovery in number-system theory was made. This was directed toward overcoming the second essential lack of the binary number system, the problem of zero redundancy. In 1957 the American mathematician George Bergman had introduced the positional number system [8]

$$A = \sum_i a_i \tau^i \tag{3}$$

where A is some real number and a_i is the ith digit binary numeral, 0 or 1, $i = 0, \pm 1, \pm 2, \pm 3$, τ^i is the weight of the ith digit, τ is the base of the number system (3).

At the first glance, there does not appear to be a difference between formulas (3) and (1), but it is only at the first glance. A principal distinction between the number system of (3) and the number system of (1) and other positional number systems, consists of the fact that Bergman used the irrational number $\tau = \frac{1}{2}(1 + \sqrt{5})$, called the 'golden ratio' [9], as the base of his number system. That is why Bergman called it the 'number system with an irrational base' or the 'Tau System'. Although Bergman's paper [8] contained a result of great importance for number system theory, it was, however, simply not noted at the time either by mathematicians or engineers; in the conclusion of his paper [8] Bergman wrote: 'I do not know of any useful application for systems such as this, except as a mental exercise and pastime, though it may be of some service in algebraic number theory'.

However, the development of computers cut across Bergman's pessimistic opinion about the practical application of his number system. In the 1970s and 1980s scientific and engineering developments based on the redundant Bergman's number system were realized in the former Soviet Union [10, 11, 12]. These developments showed the convincing effectiveness of Bergman's number system for the design of self-correcting analog-to-digital converters (ADC) and noise-tolerant processors. A theoretical basis for these developments is given in the author's monograph [10], and engineering developments are given in [11].

The main purpose of the present article is to present the scientific results of the development of a new redundant number system, the 'ternary mirror-symmetrical number system', which is a synthesis of the ternary symmetrical number system (1) and Bergman's number system (3) and which allows the solution of two problems of computer design, the sign problem and the redundancy problem.

2. THE Z-PROPERTY OF NATURAL NUMBERS

2.1. Some unusual properties of the Tau System

To understand the importance of Bergman's number system for computer science and number theory let us consider now the Tau System from the computational point of view. Its base τ determines a range of unusual properties of the number system (3). We also have to consider some mathematical properties of the golden ratio [9]. The latter is the positive root of the algebraic equation

$$x^2 = x + 1. \tag{4}$$

From (4) we can get the following identity, which connects three successive powers of the golden ratio:

$$\tau^n = \tau^{n-1} + \tau^{n-2}. \tag{5}$$

In the 19th century the French mathematician Binet proved the following mathematical formula, which allows representing the nth power of the golden ratio τ to be represented in the 'explicit' form [9]:

$$\tau^n = \frac{L_n + F_n\sqrt{5}}{2} \tag{6}$$

where n is an integer ($n = 0, \pm 1, \pm 2, \pm 3, \ldots$).

Two interesting numerical sequences appear in this formula, the Fibonacci numbers F_n and the Lucas numbers L_n. They are given with the recurrent formulae

$$F_n = F_{n-1} + F_{n-2}; \quad F_1 = F_2 = 1$$

and

$$L_n = L_{n-1} + L_{n-2}; \quad L_1 = 1; L_2 = 3.$$

Table 1 gives these sequences extended to negative values of indices.

Let us consider representations of numbers in the Tau System (3). It is clear that the abridged representation of number A given with (3) has the following form:

$$A = a_n a_{n-1} \ldots a_1 a_0, a_{-1} a_{-2} \ldots a_{-m}. \tag{7}$$

TABLE 1.

	n								
	0	1	2	3	4	5	6	7	8
F_n	0	1	1	2	3	5	8	13	21
F_{-n}	0	1	−1	2	−3	5	−8	13	−21
L_n	2	1	3	4	7	11	18	29	47
L_{-n}	2	−1	3	−4	7	−11	18	−29	47

The representation (3) is called *golden representation* of A.

We can see that the golden representation of A is some binary code combination, which is separated with a comma into two parts, the left-hand part $a_n a_{n-1} \ldots a_1 a_0$ corresponding to weights $\tau^n, \tau^{n-1}, \ldots, \tau^1, \tau^0 = 1$ and the right-hand part $a_{-1} a_{-2} \ldots a_{-m}$ corresponding to the weights with negative powers: $\tau^{-1}, \tau^{-2}, \ldots, \tau^{-m}$. Note that the weights τ^i $(i = 0, \pm 1, \pm 2, \pm 3, \ldots)$ are given by the mathematical formula (5).

The Tau System (3) overturns our traditional ideas on number system and has the following unusual properties [8, 10].

(1) One may represent in (3) some irrational numbers (the powers of the golden ratio τ^i and their sums) by using finite numbers of bits that are impossible in classical number systems (decimal, binary etc.). For example, the golden representation 100101 is the abridged notation of the irrational number $8 + 3\sqrt{5}$. The base τ of the Tau System is represented in the traditional manner

$$\tau = \frac{1 + \sqrt{5}}{2} = 10.$$

(2) The golden representations of natural numbers take the form of fractional numbers and consist of a finite number of bits. Table 2 gives golden representations of some natural numbers in (3).

(3) Each non-zero number have many golden representations in (3) and here different golden representations of the same real number A can be derived one from another by using the special transformations *convolution* and *devolution* carried out on the initial golden representation. These special code transformations are based on the identity (5). Devolution is defined as the following transformation carried out on the three neighboring digits of the initial golden representation

$$100 \to 011. \tag{8}$$

For example we can perform the following devolutions over the golden representation of number

$$5 = \mathbf{100}0,\mathbf{100}1 = 01\mathbf{10},\mathbf{0}111 = 0101{,}1111 \text{ (‘maximal form’)}. \tag{9}$$

Convolution is a back transformation, that is,

$$011 \to 100. \tag{10}$$

TABLE 2.

N	$\sum_i a_i \tau^i$	Abridged notation
1	τ^0	1,0
2	$\tau^1 + \tau^{-2}$	10,01
3	$\tau^2 + \tau^{-2}$	100,01
4	$\tau^2 + \tau^0 + \tau^{-2}$	101,01
5	$\tau^3 + \tau^{-1} + \tau^{-4}$	1000,1001
6	$\tau^3 + \tau^1 + \tau^{-4}$	1010,0001
7	$\tau^4 + \tau^{-4}$	10000,0001
8	$\tau^4 + \tau^0 + \tau^{-4}$	10001,0001
9	$\tau^4 + \tau^1 + \tau^{-2} + \tau^{-4}$	10010,0101
10	$\tau^4 + \tau^2 + \tau^{-2} + \tau^{-4}$	10100,0101

For example we can perform the following convolutions over the golden representation 0101,1111 taken from the example (9):

$$5 = 01\mathbf{01},\mathbf{1}111 = \mathbf{011}0,\mathbf{011}1 = 1000{,}1001 \text{ (‘minimal form’)}. \tag{11}$$

The notions of convolution and devolution form the basis of two special golden representations called the ‘*minimal form*’ and the ‘*maximal form*’ [10]. If we perform all possible convolutions in the golden representation of number A we will come to the minimal form of A (see the example (11)); if we perform all possible devolutions in the golden representations of A we will come to the maximal form (see the example (9)). Note that the minimal and maximal forms are the extreme golden representations possessing the following specific properties, namely:

(a) in the minimal form there are no contiguous 1s.
(b) in the maximal form there are no contiguous 0s.

Thus Bergman's number system possesses a *code redundancy*, which shows itself in an availability of many golden representations of the same number A and in the existence of two extreme golden representations for each number A, the minimal form and the maximal form. This can be used for checking informational processes in processors and computers.

Note that the notions of convolution, devolution, minimal and maximal forms are the basic notions of the redundant golden arithmetic stated in [8, 10, 11, 12].

2.2. The Z-property and the D-property

The Z-property of natural numbers [12] is the most important Tau System property having a significance for number theory.

Let us consider the representation of the arbitrary natural number of N in the Tau System:

$$N = \sum_i a_i \tau^i. \tag{12}$$

The representation of (12) is called the *τ-representation of natural number N.*

Let us apply Binet's formula (6) to the τ-representation of (12) and represent the formula (12) in the form

$$2N = \sum_i a_i L_i + \sqrt{5} \sum_i a_i F_i. \tag{13}$$

We can see that the left-hand part of expression (13) is a natural number which is always even. The right-hand part of (13) is the sum of two components. The first is the sum of the Lucas numbers with binary coefficients. Because an arbitrary Lucas number is an integer (see Table 1) then an arbitrary sum of Lucas numbers is an integer.

Let us consider now the second component. This is the product of the irrational number $\sqrt{5}$ by the sum of Fibonacci numbers with binary coefficients 0 or 1. Because an arbitrary Fibonacci number is an integer (see Table 1) then an arbitrary sum of Fibonacci numbers is always an integer. Thus, the identity (13) claims that the natural (even) number $2N$ is equal to the sum of some integer (the sum of a Lucas number with binary coefficients) and the product of the other integer (the sum of Fibonacci numbers with binary coefficients) and the irrational number $\sqrt{5}$.

There arises the question: for what conditions is the identity (13) true for arbitrary natural number N? The answer is unique. This is possible under the following condition: that the sum of Fibonacci numbers in the identity (15) equals 0 identically, that is

$$\sum_i a_i F_i = 0. \tag{14}$$

On the other hand, the number $2N$ is even. If we take into consideration the identity (14) then the sum of Lucas numbers in the identity (13) has to be always even.

Let us analyze the sum of $\sum_i a_i L_i$ and $\sum_i a_i F_i$ in the identity (13). These sums were arrived at as a result of the substitution of Binet's formula (6) into expression (12), which gives the representation of natural number N in the Tau System. This means that all binary numerals a_i in the sums of $\sum_i a_i L_i$ and $\sum_i a_i F_i$ coincide with the corresponding coefficients a_i in the τ-representation (12).

Thus as a result of this consideration we came to the following mathematical discovery in number theory.

DEFINITION. (Zero-property) *If we represent an arbitrary natural number N in the Tau System (12) and then we replace in expression (12) all powers of the golden ratio by the corresponding Fibonacci numbers, then the sum obtained in this manner has to be equal to zero identically independently from the initial natural number N.*

DEFINITION. (Double-property) *If we represent an arbitrary natural number N in the Tau System (12) and then we replace in expression (12) all powers of the golden ratio by corresponding Lucas numbers then the sum obtained in this manner has to be an even number, which is equal to double the value of the initial natural number N.*

These unique properties of the Tau System are proved in [12] and called respectively the Z-property and the D-property.

2.3. *F*- and *L*-representations

Of what practical importance could these new fundamental properties of natural numbers, Z-property and D-property, have? To answer this question let us imagine some hypothetical computer network, which uses the Tau System for number representation, in which the Z-property and the D-property formulated above are none other than a *new universal check indication on all information in the computer network.*

One can get two new representations of natural number N if we use the τ-representation (12) and the Z-property (14). Taking into consideration the Z-property (14), expression (13) can be represented in the form

$$2N = \sum_i a_i L_i + \sum_i a_i F_i$$

or

$$N = \sum_i a_i \frac{L_i + F_i}{2} = \sum_i a_i F_{i+1} \tag{15}$$

since $F_{i+1} = (L_i + F_i)/2$ (see for instance [9]). Expression (15) is called the *F-representation of natural number N* [12].

Note that the corresponding binary numerals in the τ-representation (12) and in the F-representation (15) coincide. This means that the F-representation (15) can be derived from the τ-representation of the same natural number N by simply replacing the golden ratio power τ^i in the formula of (12) by Fibonacci number F_{i+1}, where $i = 0, \pm 1, \pm 2, \pm 3, \ldots$.

Let us present now the F-representation (15) in the form

$$N = \sum_i a_i F_{i+1} + 2 \sum_i a_i F_i = \sum_i a_i L_{i+1} \tag{16}$$

since $L_{i+1} = F_{i+1} + 2F_i$ (see [9]). Expression (16) is called the *L-representation of natural number N* [12].

Note that the binary numerals in the representations (12) and (16) coincide. There follows from this that the L-representation of N could be derived from the τ-representation of the same natural number N by simply replacing the golden ratio power τ^i in the τ-representation (12) by the Lucas number L_{i+1}, where $i = 0, \pm 1, \pm 2, \pm 3, \ldots$.

It is clear that the L-representation of N could be derived also from the F-representation of the same number N by simply replacing the Fibonacci number F_{i+1} in the F-representation of the same natural number N by the Lucas number L_{i+1}.

3. TERNARY MIRROR-SYMMETRICAL REPRESENTATION

3.1. Conversion of binary golden representation to ternary golden representation

Let us consider the τ-representation of (12) for the case of its minimal form. This means that each binary unit $a_k = 1$ in the golden representation (7) would be 'enclosed' with the next two binary zeros $a_{k-1} = a_{k+1} = 0$.

Let us consider now the following expression for the powers of the golden ratio,

$$\tau^k = \tau^{k+1} - \tau^{k-1}. \tag{17}$$

Expression (17) has the following code interpretation:

$$\begin{array}{ccccccc} k+1 & k & k-1 & & k-1 & k & k-1 \\ 0 & 1 & 0 & = & 1 & 0 & \bar{1} \\ \uparrow & \downarrow & \uparrow & & & & \end{array} \tag{18}$$

where $\bar{1}$ is a negative unit, i.e. $\bar{1} = -1$. There follows from (18) that the positive binary 1 of the kth digit is transformed into two 1s, the positive 1 of the $(k+1)$th digit and the negative $\bar{1}$ of the $(k-1)$th digit.

The code transformation (18) might be used for conversion of the minimal form τ-representation (12) into the *ternary τ-representation*.

Let us convert the minimal form τ-representation of number

$$\begin{array}{cccccccccccc} & & 4 & 3 & 2 & 1 & 0 & -1 & -2 & -3 & -4 \\ 5 & = & 0 & 1 & 0 & 0 & 0, & 1 & 0 & 0 & 1 \\ & & \uparrow & \downarrow & \uparrow & & \uparrow & \downarrow & \uparrow & & \end{array} \tag{19}$$

into the ternary τ-representation. To do this we apply the code transformation (18) simultaneously to all digits that are the binary 1s and have the odd indices ($k = 2m + 1$). We can see that the transformation (18) could be applied for situation (19) only to the 3rd and (-1)th digits, being the binary 1s. As the result of such a transformation of (19) we get the following ternary representation of number 5:

$$\begin{array}{ccccccccccc} & & 4 & 3 & 2 & 1 & 0 & -1 & -2 & -3 & -4 \\ 5 & = & 1 & 0 & \bar{1} & 0 & 1, & 0 & \bar{1} & 0 & 1 \end{array}. \tag{20}$$

We can see from (20) that all digits having even indices are equal to 0 identically but the digits with odd indices take the ternary values from the set $\{\bar{1}, 0, 1\}$. This means that all digits with even indices are 'non-informative' because their values are equal to 0 identically. Omitting in (20) all the non-informative digits we get the ternary representation of the initial number N

$$N = \sum_i b_{2i}\tau^{2i}, \tag{21}$$

where b_{2i} is the ternary numeral of the $(2i)$th digit.

Let us introduce the following digit enumeration for the ternary representation (21). Each ternary digit b_{2i} is replaced with the ternary digit c_i. As a result of such enumeration we get the expression (21) in the form

$$N = \sum_i c_i\tau^{2i}, \tag{22}$$

where c_i is the ith digit ternary numeral, τ^{2i} is the weight of the ith digit and τ^2 is the base of the number system (22). With regard to expression (22) the ternary representation (20) takes the following form:

$$\begin{array}{ccccccc} & & 2 & 1 & 0 & -1 & -2 \\ 5 & = & 1 & \bar{1} & 1, & \bar{1} & 1 \end{array}. \tag{23}$$

TABLE 3.

a_{2i+1}	a_{2i}	a_{2i-1}	c_i
0	0	0	0
0	0	1	1
0	1	0	1
1	0	0	$\bar{1}$
1	0	1	0

This is the ternary τ-representation of number 5.

The conversion of the τ-representation (12) to the ternary τ-representation (22) could be performed [12, 13] by means of a simple combinative logical circuit, which transforms the next three binary digits $a_{2i+1}a_{2i}a_{2i-1}$ of the initial binary τ-representation (minimal form) into the ternary informative digit $b_{2i} = c_i$ of the ternary τ-representation in accordance with Table 3.

Note that Table 3 uses only five binary code combinations out of eight because the initial binary τ-representation has a minimal form and the code combinations 011, 110, 111 are prohibited for it.

The code transformations given with the 1st and 3rd rows of Table 3 are trivial. The code transformations given with the 2nd, 4th and 5th rows of Table 3 follow from (18). For instance, the code transformation of the 5th row $101 \Rightarrow 0$ means that the negative 1 ($\bar{1}$) arising in accordance with (18) from the left-hand binary digit $a_{2i+1} = 1$ is summarized with the positive 1 arising from the right-hand binary digit $a_{2i-1} = 1$. It follows that their sum is equal to $c_i = 0$.

Let us consider now the ternary F- and L-representations following from the ternary τ-representation (22). It is clear that the expressions for the ternary F- and L-representations, respectively, have the following forms

$$N = \sum_i c_i F_{2i+1} \tag{24}$$

and

$$N = \sum_i c_i L_{2i+1}. \tag{25}$$

Note that the values of the ternary digits in the representations (22), (24) and (25) coincide. There follows from this consideration that the ternary τ-, F- and L-representations of number 5 from the example (23) have the following numerical interpretations.

(a) Ternary τ-representation

$$\begin{aligned} 5 &= 1 \times \tau^4 + \bar{1} \times \tau^2 + 1 \times \tau^0 + \bar{1} \times \tau^{-2} + 1 \times \tau^{-4} \\ &= \frac{L_4 + F_4\sqrt{5}}{2} - \frac{L_2 + F_2\sqrt{5}}{2} + 1 \\ &\quad - \frac{L_{-2} + F_{-2}\sqrt{5}}{2} + \frac{L_{-4} + F_{-4}\sqrt{5}}{2} \\ &= L_4 - L_2 + 1 = 7 - 3 + 1. \end{aligned}$$

TABLE 4.

i	3	2	1	0	−1	−2	−3
τ^{2i}	τ^6	τ^4	τ^2	τ^0	τ^{-2}	τ^{-4}	τ^{-6}
F_{2i+1}	13	5	2	1	1	2	5
L_{2i+1}	29	11	4	1	−1	−4	−11
N							
0	0	0	0	0,	0	0	0
1	0	0	0	1,	0	0	0
2	0	0	1	$\bar{1}$,	1	0	0
3	0	0	1	0,	1	0	0
4	0	0	1	1,	1	0	0
5	0	1	$\bar{1}$	1,	$\bar{1}$	1	0
6	0	1	0	$\bar{1}$,	0	1	0
7	0	1	0	0,	0	1	0
8	0	1	0	1,	0	1	0
9	0	1	1	$\bar{1}$,	1	1	0
10	0	1	1	0,	1	1	0

(b) Ternary F-representation

$$5 = 1 \times F_5 + \bar{1} \times F_3 + 1 \times F_1 + \bar{1} \times F_{-1} + 1 \times F_{-3} = 5 - 2 + 1 - 1 + 2.$$

(c) Ternary L-representation

$$5 = 1 \times L_5 + \bar{1} \times L_3 + 1 \times L_1 + \bar{1} \times L_{-1} + 1 \times L_{-3} = 11 - 4 + 1 + 1 - 4.$$

3.2. Representation of negative numbers

Like the ternary symmetrical number system (1) the important advantage of the number system (22) is a possibility of representing both positive and negative numbers in direct code. The code of negative number $(-N)$ is derived from the ternary τ-representation of the initial number N by means of application of the rule of a ternary inversion (2). Applying this rule to the ternary τ-representation (23) of number 5 we get the ternary τ-representation of negative number (-5):

$$-5 = \begin{array}{ccccc} 2 & 1 & 0 & -1 & -2 \\ \bar{1} & 1 & \bar{1}, & 1 & \bar{1} \end{array}.$$

3.3. Mirror-symmetrical property of integer representation

Considering the ternary τ-representation (23) of number 5 we can reveal that the left-hand part $(1\bar{1})$ of the number 5 ternary τ-representation (23) is mirror-symmetrical to its right-hand part $(\bar{1}1)$ with regard to the zeroth digit. It is proved in [12] that this property of the mirror symmetry is a fundamental integer property, which arises at their representation in the Ternary Tau System (22). Table 4 demonstrates this property for some natural numbers.

Thus thanks to this simple investigation we have discovered one more fundamental property of integers, the *property of 'mirror symmetry'*, which appears at their representation in the Ternary Tau System (22). That is why the Ternary Tau System (22) is called the *Ternary Mirror-symmetrical Number System* [12].

3.4. The base of the 'Ternary Tau System'

It follows from (22) that the base of the ternary τ-representation (22) is the square of the 'golden ratio':

$$\tau^2 = \frac{3+\sqrt{5}}{2} \approx 2.618.$$

This means that the number system (22) is a number system with an irrational base. The base of the number system has the traditional representation

$$\tau^2 = 10.$$

3.5. Range of number representation

Let us consider the range of number representation in the Ternary Tau System (22). Suppose that the ternary τ-representation (22) has $2m + 1$ ternary digits. In this case one may represent in the number system (22) all integers in the range from

$$N_{\max} = \underbrace{1\,1\ldots1}_{m}\quad 1,\underbrace{1\,1\ldots1}_{m} \tag{26}$$

to

$$N_{\min} = \underbrace{\bar{1}\,\bar{1}\ldots\bar{1}}_{m}\quad \bar{1},\underbrace{\bar{1}\,\bar{1}\ldots\bar{1}}_{m}. \tag{27}$$

It is clear that $N_{\min}$ is the ternary inversion of $N_{\max}$, i.e.

$$|N_{\min}| = N_{\max}.$$

It follows from this that using the $(2m + 1)$-digit ternary τ-representation (22) one may represent

$$2N_{\max} + 1 \tag{28}$$

integers including the number 0.

For calculation of $N_{\max}$ let us use the L-representation (25). Then we can interpret the code representation (26) as the abridged notation of the sum

$$N_{\max} = L_{2m+1} + L_{2m-1} + \ldots + L_3 + L_1 + L_{-1} + L_{-3} + \ldots + L_{-2m+1}. \tag{29}$$

For the odd indices $i = 2k - 1$ we have the following property for Lucas numbers [9]:

$$L_{-2m+1} = -L_{2m-1}. \tag{30}$$

Taking into consideration the property (30) we get the following value of the sum (29):

$$N_{\max} = L_{2m+1}. \tag{31}$$

Taking into consideration (28) and (31) one may formulate the following theorem.

THEOREM 1. *Using* $(2m+1)$ *ternary digits in the ternary* τ*-representation (22) one may represent* $2L_{2m+1}+1$ *integers in the range from* $-L_{2m+1}$ *to* L_{2m+1}*, where* L_{2m+1} *is a Lucas number.*

3.6. Redundancy of the ternary mirror-symmetrical representation

Let us compare the number system (22) with the ternary symmetrical number system (1).

It is well known [6] that one may represent the 3^n integers by using the n-digit ternary symmetrical number system (1) in the range from $N_{\min} = -(3^n - 1)/2$ up to $N_{\max} = (3^n - 1)/2$.

For representation of the number range given with Theorem 1 by using the n-digit ternary symmetrical number system (1) it is necessary to fulfill the inequality

$$3^n \geq 2L_{2m+1} + 1 \approx 2L_{2m+1}. \tag{32}$$

The following formula is well known from Fibonacci numbers theory [9], which allows us to express Lucas number L_{2m+1} through the golden ratio power

$$L_{2m+1} \approx \tau^{2m+1}. \tag{33}$$

Inserting (33) into (32) and taking the logarithm by the radix 3 of both parts of inequality (32) we can get the inequality

$$n \geq (2m + 1) \log_3 \tau + \log_3 2. \tag{34}$$

With an increase of m the inequality (34) turns out to be the following approximate equality:

$$n \approx (2m + 1) \log_3 \tau. \tag{35}$$

The equality (35) can be used for calculation of code redundancy of the ternary mirror-symmetrical number system. A relative redundancy R is calculated through the formula

$$R = \frac{k - n}{n} = \frac{k}{n} - 1,$$

where k and n are the digit numbers of the redundant and non-redundant number system for representation of the same number range.

If we choose $k = 2m + 1$ digits for the ternary mirror-symmetrical number representation then we will need n digits for the representation of the same number range in number system (1). From this consideration there follows the following expression for calculation of the relative code redundancy of the ternary mirror-symmetrical number system:

$$R = \frac{1 - \log_3 \tau}{\log_3 \tau} = 1.283 = 128.3\%. \tag{36}$$

Thus the relative redundancy of the ternary mirror-symmetrical number system is sufficiently large. However, this deficiency is compensated by some interesting possibilities, such as checking golden number representations and arithmetical operations over them.

TABLE 5.

	a_k		
b_k	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}1\bar{1}$	$\bar{1}$	0
0	$\bar{1}$	0	1
1	0	1	$1\bar{1}1$

4. TERNARY MIRROR-SYMMETRICAL ARITHMETIC

4.1. Mirror-symmetrical addition and subtraction

The following identities for the golden ratio powers underlie mirror-symmetrical addition:

$$2\tau^{2k} = \tau^{2(k+1)} - \tau^{2k} + \tau^{2(k-1)}, \tag{37}$$

$$3\tau^{2k} = \tau^{2(k+1)} + 0 + \tau^{2(k-1)}, \tag{38}$$

$$4\tau^{2k} = \tau^{2(k+1)} + \tau^{2k} + \tau^{2(k-1)}, \tag{39}$$

where $k = 0, \pm 1, \pm 2, \pm 3, \ldots$.

The identity (37) is a mathematical basis for the mirror-symmetrical addition of two single-digit ternary digits and gives the rule of carry realization (Table 5).

A principal peculiarity of Table 5 is the addition rule for two ternary 1s with equal signs, i.e.

$$\begin{array}{ccccccc} a_k & + & b_k & = & c_k & s_k & c_k \\ 1 & + & 1 & = & 1 & \bar{1} & 1 \\ \bar{1} & + & \bar{1} & = & \bar{1} & 1 & \bar{1} \end{array} .$$

We can see that for the mirror-symmetrical addition of the ternary 1s with the same sign there arises the intermediate sum s_k with the opposite sign and the carry c_k with the same sign. However the carry from the kth digit spreads simultaneously to the next two digits, namely to the next left-hand, i.e. $(k + 1)$th digit, and to the next right-hand, i.e. $(k - 1)$th digit.

Table 5 describes an operation of the simplest adder called a *single-digit half-adder*. The latter is a combinative logical circuit having the two ternary inputs a_k and b_k and the two ternary outputs s_k and c_k and functioning in accordance with Table 5 (Figure 1a).

Since the carry from the kth digit spreads to the left-hand and to the right-hand digits this means that the full mirror-symmetrical single-digit adder has to have two inputs for the carries entering from the $(k - 1)$th and $(k + 1)$th digits into kth digit. Thus the full mirror-symmetrical single-digit adder is a combinative logical circuit having four ternary inputs and two ternary outputs (Figure 1b). Let us denote as 2Σ the mirror-symmetrical single-digit half-adder having two inputs and as 4Σ the mirror-symmetrical single-digit full adder having four inputs.

Let us describe the logical operation of the mirror-symmetrical full single-digit adder 4Σ. Note first of all that the number of all the possible four-digit ternary input

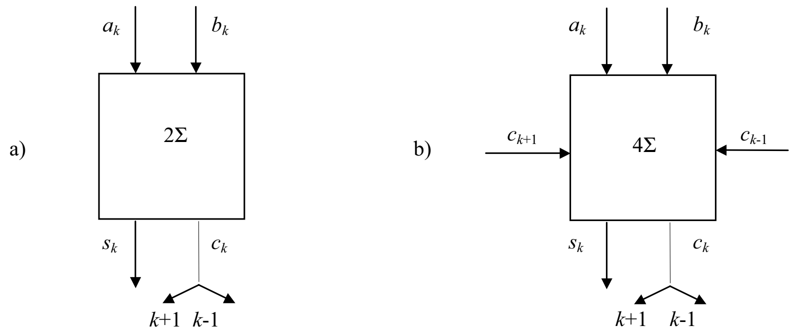

FIGURE 1. Mirror-symmetrical single-digit adders: (a) a half-adder; (b) a full adder.

combinations of the mirror-symmetrical full adder equals $3^4 = 81$. The values of the output variables s_k and c_k are some discrete functions of the algebraic sum S of the input ternary variables $a_k, b_k, c_{k-1}, c_{k+1}$, i.e.

$$S = a_k + b_k + c_{k-1} + c_{k+1}. \tag{40}$$

The sum (40) takes the values from the set $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$. The operation rule of the mirror-symmetrical full adder 4Σ consists of the following. The adder forms the output ternary code combination $c_k s_k$ in accordance with the value of the sum (40), i.e.

$$-4 = \bar{1}\,\bar{1}; \quad -3 = \bar{1}\,0; \quad -2 = \bar{1}\,1; \quad -1 = 0\,\bar{1};$$
$$0 = 00; \quad 1 = 01; \quad 2 = 1\,\bar{1}; \quad 3 = 10; \quad 4 = 11.$$

The lower digit of such a two-digit ternary representation is the value of the intermediate sum s_k, and the higher digit is the value of the carry c_k, which spreads to the next (to the left-hand and to the right-hand) digits.

The multi-digit combinative mirror-symmetrical adder realizing the addition of two $(2m + 1)$-digit mirror-symmetrical numbers is a combinative logical circuit consisting of $2m + 1$ ternary mirror-symmetrical full single-digit adders 4Σ (Figure 2).

As an example let us consider the addition of two numbers $5 + 10$ in the Ternary Tau System:

5	=	0	1	$\bar{1}$	1,	$\bar{1}$	1	0
10	=	0	1	1	0,	1	1	0
		0	$\bar{1}$	0	1,	0	$\bar{1}$	0
		1	$\leftrightarrow$	1		1	$\leftrightarrow$	1
15	=	1	$\bar{1}$	1	1,	1	$\bar{1}$	1

Note the sign $\leftrightarrow$ marks the process of carry spreading.

As was noted above the important advantage of the Ternary Tau System lies in the possibility of summarizing all integers (positive and negative) in direct code, i.e. without using notions of inverse and additional codes. As an example let us consider the addition of negative number (-24) with positive number 15:

−24	=	$\bar{1}$	$\bar{1}$	0	1,	0	$\bar{1}$	$\bar{1}$
15	=	1	$\bar{1}$	1	1,	1	$\bar{1}$	1
		0	1	1	$\bar{1}$,	1	1	0
			$\downarrow$	1	$\leftrightarrow$	1	$\downarrow$	
		$\bar{1}$	$\leftrightarrow$	$\bar{1}$		$\bar{1}$	$\leftrightarrow$	$\bar{1}$
−9	=	$\bar{1}$	1	1	$\bar{1}$,	1	1	$\bar{1}$

Subtraction of two mirror-symmetrical numbers $N_1 - N_2$ is reduced to addition if we represent their difference in the form

$$N_1 - N_2 = N_1 + (-N_2). \tag{41}$$

It follows from (41) that before subtraction it is necessary to take the ternary inversion of number N_2.

4.2. 'Swing' phenomenon

Let us sum two equal numbers $5 + 5$ represented in the Ternary Tau System:

5	=	0	1	$\bar{1}$	1,	$\bar{1}$	1	0
5	=	0	1	$\bar{1}$	1,	$\bar{1}$	1	0
		0	$\bar{1}$	1	$\bar{1}$,	1	$\bar{1}$	0
			$\downarrow$	1	$\leftrightarrow$	1	$\downarrow$	
		1	$\leftrightarrow$	1		1	$\leftrightarrow$	1
			$\bar{1}$	$\leftrightarrow$	$\bar{1}$	$\downarrow$		
					$\bar{1}$	$\leftrightarrow$	$\bar{1}$	
		1	1	0	0,	0	1	1
				$\bar{1}$	$\leftrightarrow$	$\bar{1}$		
			1	$\leftrightarrow$	1	$\downarrow$		
			$\downarrow$		1	$\leftrightarrow$	1	
		$\bar{1}$	$\leftrightarrow$	$\bar{1}$		$\bar{1}$	$\leftrightarrow$	$\bar{1}$
		0	$\bar{1}$	1	$\bar{1}$,	1	$\bar{1}$	0
				1	$\leftrightarrow$	1		
			$\bar{1}$	$\leftrightarrow$	$\bar{1}$	$\downarrow$		
			$\downarrow$		$\bar{1}$	$\leftrightarrow$	$\bar{1}$	
		1	$\leftrightarrow$	1		1	$\leftrightarrow$	1

It follows from this example that we have found a special addition case called a *swing*. If the addition process continues, starting from some step, the process of the carry

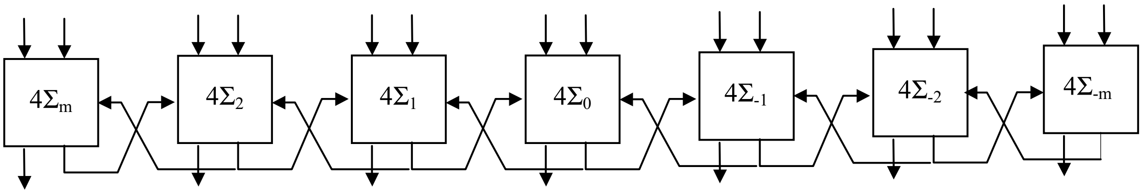

FIGURE 2. Ternary mirror-symmetrical multi-digit adder.

formation turns out to be repetitive and hence the addition becomes infinite. The swing phenomenon is a variety of the 'races' arising in digit automatons when elements are switching over.

To eliminate the swing phenomenon one may use the following effective 'technical' method [12]. Let us delay the input signals of single-digit adders with odd indices ($k = \pm 1, \pm 3, \pm 5, \ldots$) by one addition step. With this aim in mind the adders with the even indices ($k = 0, \pm 2, \pm 4, \ldots$) form at the first addition step intermediate sums and corresponding carries to the single-digit adders with the odd indices. At the second addition step the carries formed at the first step are summed with the corresponding ternary variables of the odd digits of summed numbers. Thanks to such an approach the swing phenomenon is eliminated.

Let us demonstrate this method for the preceding example of adding $5 + 5$:

$$\begin{array}{cccccccccc}
5 & = & 0 & 1 & \bar{1} & 1, & \bar{1} & 1 & 0 \\
5 & = & 0 & 1 & \bar{1} & 1, & \bar{1} & 1 & 0 \\
\hline
 & & 0 & \bar{1} & \bar{1} & \bar{1}, & \bar{1} & \bar{1} & 0 \\
 & & 0 & & \bar{1} & \downarrow & \bar{1} & & 0 \\
 & & & & 1 & \leftrightarrow & 1 & & \\
 & & 1 & \leftrightarrow & 1 & & 1 & \leftrightarrow & 1 \\
\hline
10 & = & 1 & \bar{1} & 0 & \bar{1}, & 0 & \bar{1} & 1
\end{array}$$

The first step of mirror-symmetrical addition is carry formation from all the digits with even indices (0, 2, −2). The numerals of all the input digits with odd indices (1, 3, −1, −3) are preserved without exchange at the second step. The second step is addition of all carries arising at the first step with the intermediate sum and the numerals of the digits with the odd indices.

The analysis of all considered examples of the ternary mirror-symmetrical addition shows that both the final addition result and all the intermediate addition results are mirror-symmetrical numbers, i.e. the property of mirror symmetry is the invariant about mirror-symmetrical addition.

Note that the adder in Figure 2 consists of two mirror-symmetrical parts in relation to the 0th single-digit adder $4\Sigma_0$. The right-hand part of the adder in Figure 2 is used for checking the output information according to the principle of mirror symmetry. There exists a possibility to decrease a 'structural redundancy' of the adder in Figure 2 if we use a 'doubling' interpretation of the ternary mirror-symmetrical adders (see Figure 3).

TABLE 6.

	a_k		
b_k	$\bar{1}$	0	1
$\bar{1}$	1	0	$\bar{1}$
0	0	0	0
1	$\bar{1}$	1	1

The ternary mirror-symmetrical adder in Figure 3 consists of two parts. Here the carry output of the single-digit adder $4\Sigma_1$ in the left-hand part of the adder is connected with the two inputs of the single-digit adder $4\Sigma_0$, and the carry output of the single-digit adder $4\Sigma_{-1}$ of the right-hand part of the adder is connected with the two inputs of the doubling single-digit adder $4\Sigma_0$. This assures correct mirror-symmetrical addition in the left-hand and right-hand parts of the adder.

This means that we can use only one part of the adder in Figure 3 for ternary mirror-symmetrical computations. At the concluding stage of computations we can restore the final ternary mirror-symmetrical representation according to the principle of mirror symmetry.

4.3. Mirror-symmetrical multiplication

The following trivial identity for the golden ratio powers underlies mirror-symmetrical multiplication:

$$\tau^{2n} \times \tau^{2m} = \tau^{2(n+m)}. \qquad (42)$$

The rule of the mirror-symmetrical multiplication of two single digits is given in Table 6.

Multiplication is performed in direct code. The general algorithm of two multi-digit mirror-symmetrical number multiplication is reduced to forming the partial products in accordance with Table 6 and their addition in accordance with the rule of the mirror-symmetrical addition. For example let us multiply the negative number $-6 = \bar{1}01{,}0\bar{1}$ by the positive number $2 = 1\bar{1}{,}1$:

$$\begin{array}{ccccccc}
 & \bar{1} & 0 & 1, & 0 & \bar{1} & \\
 & & 1 & \bar{1}, & 1 & & \\
\hline
 & & \bar{1} & 0, & 1 & 0 & \bar{1} \\
 & 1 & 0 & \bar{1}, & 0 & 1 & \\
\bar{1} & 0 & 1 & 0, & \bar{1} & & \\
\hline
\bar{1} & 1 & 0 & \bar{1}, & 0 & 1 & \bar{1}
\end{array}$$

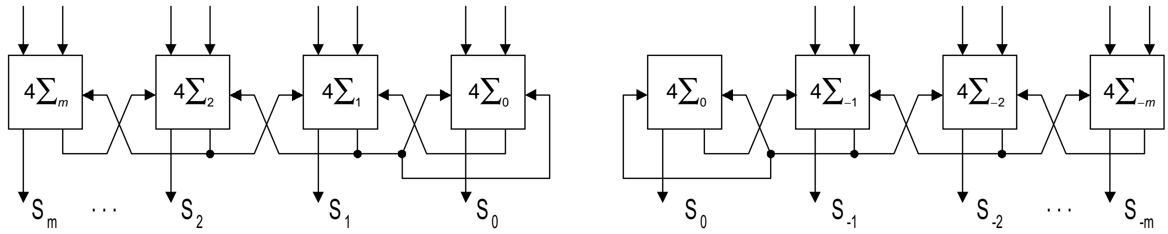

FIGURE 3. ‘Doubling’ interpretation of the ternary mirror-symmetrical adder.

The multiplication result in the above-considered example is formed as the sum of three partial products. The first partial product $\bar{1}0,10\bar{1}$ is the result of multiplication of the mirror-symmetrical number $-6 = \bar{1}01,0\bar{1}$ by the lowest positive unit of the mirror-symmetrical number $2 = 1\bar{1},1$; the second partial product $10\bar{1},01$ is the result of the multiplication of the same number $-6 = \bar{1}01,0\bar{1}$ by the middle negative unit of the number $2 = 1\bar{1},1$; and finally the third partial product $\bar{1}01,0\bar{1}$ is the result of the multiplication of the same number $-6 = \bar{1}01,0\bar{1}$ by the higher positive unit of the number $2 = 1\bar{1},1$.

Note that the product $-12 = \bar{1}\ 1\ 0\ \bar{1},\ 0\ \bar{1}\ \bar{1}$ preserves the property of mirror symmetry. Since its higher digit is a negative unit $\bar{1}$ it follows that the product is a negative number.

Mirror-symmetrical division is performed in accordance with the rule of division of the classical ternary symmetrical number system [6]. The general algorithm of the mirror-symmetrical division is reduced to the sequential subtraction of a shifted divisor, which is multiplied by the next ternary numeral of the quotient.

5. MATRIX AND PIPELINE MIRROR-SYMMETRICAL ADDERS

It is well known that digital signal processors put high demands on the speed of arithmetical devices. Different special structures (matrix, pipeline, etc.) are elaborated for this purpose. Let us show that the ternary mirror-symmetrical arithmetic contains in itself interesting possibilities for the realization of fast arithmetical processors.

Let us consider the matrix multi-digit ternary mirror-symmetrical adder (Figure 4).

Each cell of the matrix adder in Figure 4 is a ternary-symmetrical single-digit full adder having four inputs and two outputs (see Figure 1b). The matrix adder in Figure 4 consists of the 21 single-digit full adders arranged in the form of the 7 × 3-matrix. Each ternary single-digit adder has a designation $4\Sigma_3^2$ where the number 4 means that the adder has four ternary inputs, the lower index i means a digit number in the ternary mirror-symmetrical representation (22) and the higher index k means a row number of the matrix adder in Figure 4.

The inputs of the single-digit adders

$$4\Sigma_3^1, 4\Sigma_2^1, 4\Sigma_1^1, \Sigma_0^1, 4\Sigma_{-1}^1, 4\Sigma_{-2}^1, 4\Sigma_{-3}^1$$

of the first row form the multi-digit input of the matrix ternary-symmetrical adder. The output of the intermediate sum of each single-digit adder is connected to the corresponding input of the next single-digit adder of the same column.

The outputs of the intermediate sum of the single-digit adders

$$4\Sigma_3^1, 4\Sigma_2^1, 4\Sigma_1^1, \Sigma_0^1, 4\Sigma_{-1}^1, 4\Sigma_{-2}^1, 4\Sigma_{-3}^1$$

of the last row form the multi-digit output of the matrix mirror-symmetrical adder.

The basic peculiarity of the matrix mirror-symmetrical adder in Figure 4 is a special design of connections between carry outputs of the single-digit adders and the inputs of the neighboring single-digit adders. The carry outputs of all single-digit adders with the lower even indices (2, 0, −2) are connected with the corresponding inputs of the neighboring single-digit adders aligned in the same row, but the carry outputs of all single-digit adders with the lower odd indices (3, 1, −1, −3) are connected with the corresponding inputs of the neighboring single-digit adders aligned in the lower row. Note that such organization of carry connections allows the elimination of the above-considered swing phenomenon.

Let us consider an operation of the matrix mirror-symmetrical adder for a concrete example. Let us sum two equal mirror-symmetrical numbers

$$A = 0111{,}110 \quad B = 0111{,}110.$$

The addition consists of two stages. Each of the stages is realized by one row of single-digit adders and consists of two steps.

5.1. The first stage

In accordance with Figure 3 the first step consists of the following. The single-digit adders of the first row with the lower even indices ($4\Sigma_2^1$, $4\Sigma_0^1$, $4\Sigma_{-2}^1$) form intermediate sums, which enter the inputs of the second row adders, and the carries, which enter the corresponding inputs of the single-digit adders with the lower odd indices of the first row ($4\Sigma_3^1$, $4\Sigma_1^1$, $4\Sigma_{-1}^1$, $4\Sigma_{-3}^1$). The above-considered transformation of the code information may be presented in

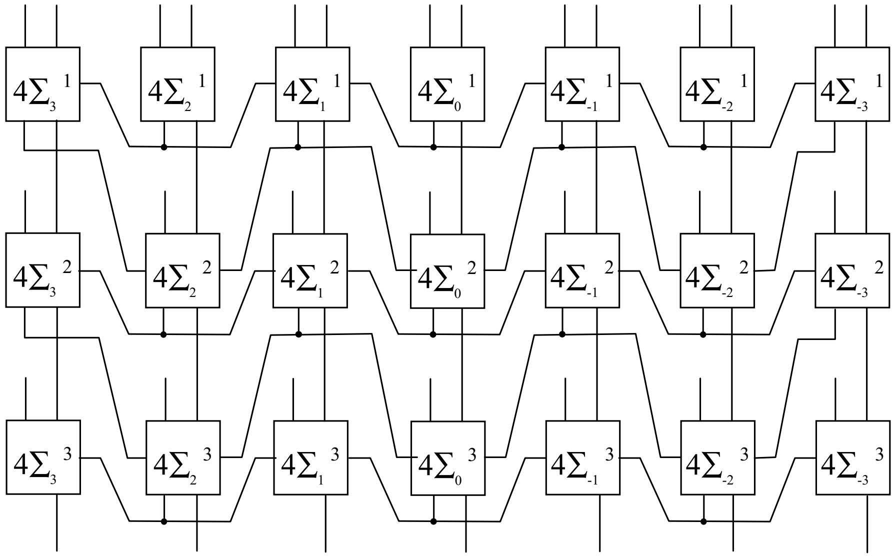

FIGURE 4. Matrix ternary mirror-symmetrical adder.

the following form:

$$
\begin{array}{ccccccc}
0 & 1 & 1 & 1, & 1 & 1 & 0 \\
0 & 1 & 1 & 1, & 1 & 1 & 0 \\
\hline
 & \bar{1} & & \bar{1} & & \bar{1} & \\
 & \downarrow & 1 & \leftrightarrow & 1 & \downarrow & \\
1 & \leftrightarrow & 1 & & 1 & \leftrightarrow & 1
\end{array}
$$

Hence the first step is a formation of the intermediate sums and the carries on the outputs of the single-digit adders with the lower even indices (2, 0, −2).

At the second step the single-digit adders with the lower odd indices (3, 1, −1, −3) enter into action. In accordance with the entered carries they form the intermediate sums and the carries entering the single-digit adders of the lower row, i.e.

$$
\begin{array}{ccccccc}
0 & 1 & 1 & 1, & 1 & 1 & 0 \\
0 & 1 & 1 & 1, & 1 & 1 & 0 \\
\hline
 & \bar{1} & & \bar{1} & & \bar{1} & \\
 & \downarrow & 1 & \leftrightarrow & 1 & \downarrow & \\
1 & \leftrightarrow & 1 & & 1 & \leftrightarrow & 1 \\
\hline
1 & \bar{1} & 1 & \bar{1}, & 1 & \bar{1} & 1 \\
 & 1 & \leftrightarrow & 1 & \downarrow & & \\
 & & & 1 & \leftrightarrow & 1 &
\end{array}
$$

The first stage is over. We can see that the results of the first stage are some intermediate sum and some carries entering the adders of the lower row.

5.2. The second stage

The single-digit adders of the second row with the lower even indices ($4\Sigma_2^2$, $4\Sigma_0^2$, $4\Sigma_{-2}^2$) form the intermediate sums entering the corresponding inputs of the lower row adders and the carries entering the corresponding inputs of the same row adders with the lower odd indices ($4\Sigma_3^2$, $4\Sigma_1^2$, $4\Sigma_{-1}^2$, $4\Sigma_{-3}^2$), i.e.

$$
\begin{array}{ccccccc}
1 & \bar{1} & 1 & \bar{1}, & 1 & \bar{1} & 1 \\
 & 1 & \leftrightarrow & 1 & \downarrow & & \\
 & & & 1 & \leftrightarrow & 1 & \\
\hline
1 & 0 & 1 & 1, & 1 & 0 & 1
\end{array}
$$

Since all the carries forming at this stage became equal to zero this means that addition is over at the second stage (this is true only for the case considered). The obtained sum enters the inputs of the lower row adders $4\Sigma_3^3 - 4\Sigma_{-3}^3$ and then appears on the output of the adder.

There exist two directions for extension of functional possibilities of the matrix mirror-symmetrical adder in Figure 4. If we set ternary registers between neighboring rows of the single-digit adders then the matrix adder considered above turns into the *pipeline mirror-symmetrical adder*. In fact the code information from preceding rows of the single-digit adders is memorized in corresponding registers, and the preceding row of the adders is ready for a further processing. Then the adders of the lower row process the code information entering the lower row single-digit adders, and simultaneously the top row of the single-digit adders starts to process the new code information. This means that, starting at a given moment, we will get the sums of numbers $A_1 + B_1, A_2 + B_2, \ldots, A_n + B_n$ entering the adder input during the time period $2\Delta\tau$, where $\Delta\tau$ is the delay time of the single-digit adder.

The other possibility of extending the functional possibilities of the pipeline adder consists of the following. We can see from Figure 4 that each single-digit adder of the lower rows has a ‘free’ input. We can use these inputs as new multi-digit outputs of the pipeline adder. Using these multi-digit inputs we can turn the pipeline adder into the pipeline multiplier. In this case mirror-symmetrical multiplication of two mirror-symmetrical numbers $A(1) \times B(1)$ is performed in the following way. The first row of the single-digit adders

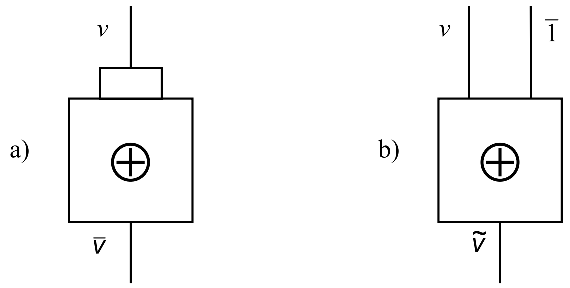

FIGURE 5. Logical circuits of ternary inversion (a) and cyclic negation (b).

summarizes the first two partial products $P_1^1 + P_2^1$. This code information enters the second row of the single-digit adders. If we send the third partial product P_3^1 to the free input of the second row we will get the sum $P_1^1 + P_2^1 + P_3^1$ on the outputs of the second row. In so doing the first row starts summing up the first two partial products of the next pair of multiplied numbers $A(2) \times B(2)$. The free input of the third row is used for receiving the next partial product P_4^1 of the first pair of the multiplied numbers $A(1) \times B(1)$, etc. We can see that the pipeline adder in Figure 4 allows the multiplication of many mirror-symmetrical numbers in the pipeline regime. In so doing the multiplication speed is determined by the time $2\Delta\tau$, where $\Delta\tau$ is the delay time of the single-digit adder.

6. TECHNICAL REALIZATION OF TERNARY MIRROR-SYMMETRICAL STRUCTURES BY USING BINARY LOGICAL ELEMENTS

6.1. Basic functions of ternary logic

Ternary logic is a special case of so-called k-valued logic ($k = 2, 3, 4, 5, \dots$) for the case $k = 3$. For coordination with the ternary-symmetrical number system let us assume that ternary logical variables take their values from the set $\{\bar{1}, 0, 1\}$.

Then the basic logical functions of one ternary variable v are determined in the following way:

Inversion function	Cyclic negation
$f(v) = \bar{v} = \begin{cases} \bar{1} \text{ with } v = 1 \\ 0 \text{ with } v = 0 \\ 1 \text{ with } v = \bar{1} \end{cases}$	$f(v) = \tilde{\tilde{v}} = \begin{cases} \bar{1} \text{ with } v = 0 \\ 0 \text{ with } v = 1 \\ 1 \text{ with } v = \bar{1} \end{cases}$

Let us consider the following important functions of *two ternary variables*.

(1) Ternary conjunction $f(v_1, v_2) = \min(v_1, v_2) = v_1 \wedge v_2$

$\wedge$	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}$	$\bar{1}$	$\bar{1}$
0	$\bar{1}$	0	0
1	$\bar{1}$	0	1

(2) Ternary disjunction $f(v_1, v_2) = \max(v_1, v_2) = v_1 \vee v_2$

$\vee$	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}$	0	1
0	0	0	1
1	1	1	1

(3) Addition by modulo 3 $f(v_1, v_2) = v_1 \oplus v_2 \pmod 3$

$\oplus$	$\bar{1}$	0	1
$\bar{1}$	1	$\bar{1}$	0
0	$\bar{1}$	0	1
1	0	1	$\bar{1}$

(4) Multiplication by modulo 3 $f(v_1, v_2) = v_1 \otimes v_2 \pmod 3$

$\otimes$	$\bar{1}$	0	1
$\bar{1}$	1	0	$\bar{1}$
0	0	0	0
1	$\bar{1}$	0	1

There exist the following identities for the above-introduced ternary logical functions:

$$\bar{\bar{v}} = v; \quad v \wedge v = v; \quad v \wedge \bar{1} = \bar{1}; \quad v \wedge 1 = v,$$
$$v \vee v = v; \quad v \vee 1 = 1; \quad v \vee \bar{1} = v,$$
$$v \oplus 0 = v; \quad v \otimes 0 = 0; \quad v \otimes 1 = v; \quad v \otimes \bar{1} = \bar{v}.$$

Ternary functions of conjunction, disjunction and inversion are connected with Morgan's formulas:

$$\overline{v_1 \wedge v_2} = \overline{v_1} \vee \overline{v_2}, \quad \overline{v_1 \vee v_2} = \overline{v_1} \wedge \overline{v_2}.$$

Note that the three-valued logic used in the Setun [2] appeared for engineers as the long known logic of positive, negative and equal-to-zero electrical current, and for programmers as the logic of number signs: $+$, $-$, 0, etc.

As in Boolean logic there exist different variants of the functionally complete systems of ternary logic. We can use so-called 'modular logic' functions for synthesizing ternary logical elements. The system of modular logic includes the functions

$$f(v_1, v_2) = v_1 \oplus v_2, \quad f(v_1, v_2) = v_1 \otimes v_2. \tag{43}$$

Let us add to modular functions (43) the special function

$$f(v_1, v_2) = v_1 \Theta v_2. \tag{44}$$

The latter gives the rule of carry formation for addition of single-digit ternary numbers. The logical table of this function has the form

Θ	$\bar{1}$	0	1
$\bar{1}$	$\bar{1}$	0	0
0	0	0	0
1	0	0	1

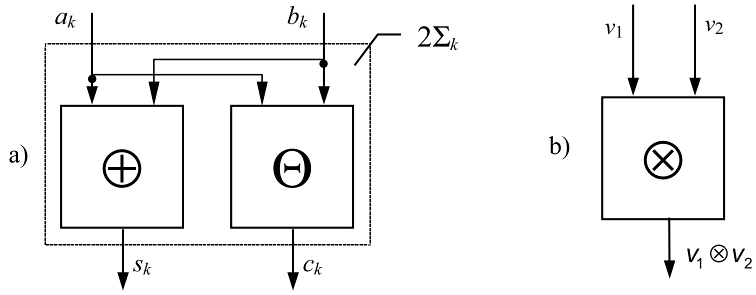

FIGURE 6. Ternary single-digit half-adder (a) and multiplier (b).

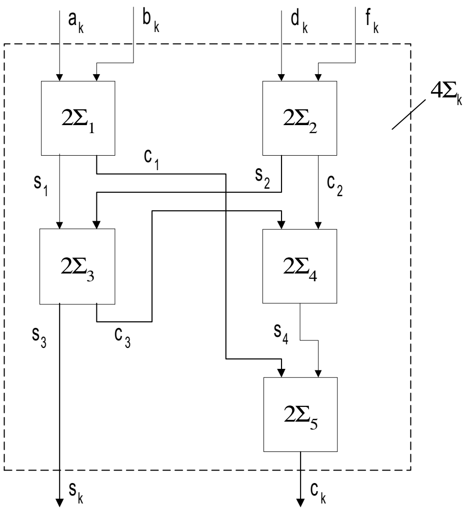

FIGURE 7. Full ternary single-digit adder.

TABLE 7.

v	x_1	x_2
$\bar{1}$	1	0
0	0	0
1	0	1

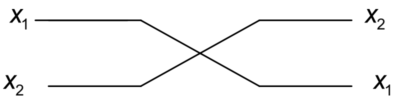

FIGURE 8. Logical circuit of ternary inversion realized by using binary variables.

The set of the ternary functions (42), (43) is a functionally complete set of the ternary logical functions, which may be used for synthesizing ternary logical circuits. It is easy to prove that the ternary inversion function $\overline{v}$ and the cyclic negation function $\overset{\approx}{v}$ are realized by using the modulo 3 addition circuit (Figure 5).

The ternary single-digit half-adder 2Σ is realized by using the logical circuits $\oplus$ and Θ (Figure 6a) and the ternary single-digit multiplier is the logical circuit $\otimes$ (Figure 6b).

If we take the ternary single-digit half-adder 2Σ as the basic logical circuit for realization of the ternary-symmetrical arithmetic devices one may prove that the five single-digit half-adders form the ternary single-digit full adder 4Σ (Figure 7).

For micro-electronic realization from the view of VLSI implementation we can use the following binary coding of the ternary variable as is shown in Table 7.

Then each ternary circuit, for instance the ternary adder in Figure 6, can be presented in VLSI having ternary inputs and outputs. All ternary variables on each ternary input are transformed into binary variables according to Table 7 and all pairs of binary variables corresponding to some ternary variable are transformed to corresponding ternary variable on corresponding output of the ternary circuit according to Table 7. Then the problem of design of ternary circuit is reduced to design of binary VLSI by using modern VLSI design methods.

Note that some ternary functions are realized very simply in this manner. For example the logical circuit of 'ternary inversion' $f(v) = \overline{v}(v = x_1x_2$ and $v = x_2x_1)$ is realized as shown in Figure 8.

6.2. Flip-flap-flop

The same binary approach can be used for design of a ternary memory element called flip-flap-flop. It is well known that the basis of a classical binary flip-flop is a logical circuit consisting of two logical elements 1 and 2 of the kind *OR–NOT* (Figure 9a), which are connected with back logical connections.

Let us consider now a logical circuit consisting of the three logical elements 1, 2, 3 of the kind OR–NOT (Figure 9b). Let us assume that the logical elements 2 and 3 are neighbors to the logical element 1, the logical elements 3 and 1 are neighbors to the logical element 2, and the logical elements 1 and 2 are neighbors to the logical element 3. Each logical element OR–NOT is connected with its neighboring logical elements with back logical connections. This gives rise to three stable states of the logical circuit in Figure 9b. In fact, let us assume that we have the logical 1 on the input C of the logical element 2. This logical 1 enters the inputs of the neighboring logical elements 1 and 3 and supports the logical 0 on their outputs A and B. These logical zeros enter the inputs of the logical element 2 and support the logical 1

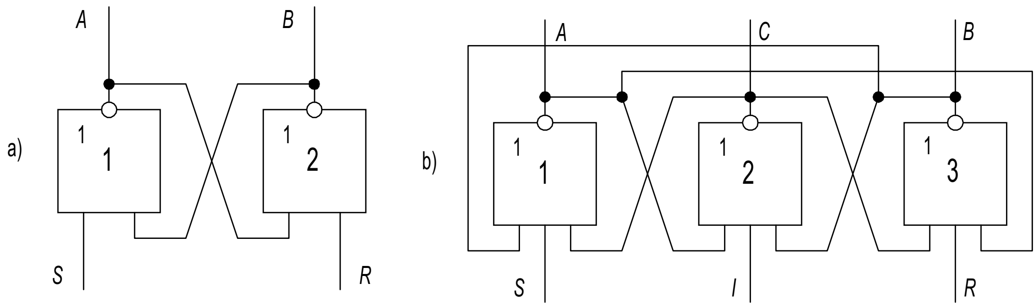

FIGURE 9. Flip-flop (a) and flip-flap-flop (b).

on its output C. Hence this state of the circuit in Figure 8b is the first stable state. This stable state corresponds to the code combination 010 on the outputs A, C, B. One may show that the circuit has one more two stable states corresponding to the code combinations 100 and 001 on the outputs A, C, B. We can use these stable states of the circuit in Figure 8b for the binary coding of the ternary numerals according to

$$0 = 010$$
$$1 = 001$$
$$\bar{1} = 100.$$

If we eliminate the middle output C we will get the binary outputs A and B, which correspond to the binary coding of ternary variables according to Table 7.

Hence the logical circuit in Figure 9b may be considered as a ternary–binary memory element called flip-flap-flop. Let us consider the functioning of the flip-flap-flop in Figure 9b. It has three stable states $\bar{1}$, 0 and 1. Let the flip-flap-flop in Figure 8b be in the state $Q = 0$. This means that the output $C = 1$, other outputs $A = B = 0$. If we need to set the flip-flap-flop into the state $Q = 1(001)$ we should send to the flip-flap-flop inputs S, I, R the following code signals $S = 1$, $I = 1$, $R = 0$. The signals $S = 1$ and $I = 1$ cause the appearance of the logical zeros on the outputs A and C. These logical zeros enter the inputs of the logical element 3 and together with the logical signal $R = 0$ cause an appearance of the logical 1 on the output B.

By analogy one may show that the signals $S = 0$, $I = 1$, $R = 1$ turn the flip-flap-flop in Figure 8b into the state $\bar{1}$ (100).

7. THE MOST EFFECTIVE FIELDS OF APPLICATIONS

7.1. Analog-to-digit and digit-to-analog converters (ADC and DAC)

The most effective application of Bergman's number system and the ternary mirror-symmetrical number system is ADC and DAC [10, 11, 12]. Since this application is stated in more detail in the author's monograph [10] and in the article collection [11] one may formulate the basic results of the golden ADC and DAC investigations [10, 11] as follows.

(1) The golden ADC and DAC have structures similar to the ADC and DAC structures based on the classical binary number system.
(2) However, Bergman's number system gives to the golden ADC and DAC a necessary code redundancy (each analog value can be presented with different golden representations). This allows us to improve all basic parameters of the golden ADC and DAC in comparison with binary ADC and DAC, in particular to increase the ADC spread by a factor of four to five and to design self-controlling and self-correcting ADC and DAC, which are 'insensitive' to an exactness of analogous ADC and DAC elements, and to the influence of temperature and other external conditions. With an error of analogous ADC and DAC of about 5% the principle of self-correction allows us to decrease this error to 0.005% and to design self-correcting 16-bit ADC and DAC functioning in a wide range of temperature change [11].

As an example let us consider the electrical realization of DAC for the ternary mirror-symmetrical representation (Figure 10).

The golden DAC in Figure 10 is intended for conversion of the ternary mirror-symmetrical representation into analogous values. This one consists of the 5 ($2m + 1$ in the general case) digits. The middle point C corresponds to the zero digit a_0 of the input golden ternary mirror-symmetrical code $a_2a_1a_0, a_{-1}a_{-2}$ ($a_ma_{m-1}\dots a_0, a_{-1}a_{-2}\dots a_{-m}$ in the general case) of number N. The ternary digits a_i ($i = 0, \pm1, \pm2, \dots, \pm m$) are controlled with the special circuit I_0 connected with corresponding connection points of the golden mirror-symmetrical resistive divisor. Note that the golden resistive divisor in Figure 10 consists of the resistors by value R and two ended resistors by value τR where τ is the golden ratio. It is unexpected that the coefficient of electrical voltage transmission in the golden resistive divisor in Figure 10 is equal to the square of the golden ratio [12].

The special circuit I_0 in Figure 10 consists of the standard electrical generator I_0 and the 3-position electrical key, which are controlled by the ternary digits a_i according to the following rule. If $a_i = 1$, then the standard electrical current is switched to the corresponding point of the golden mirror-symmetrical resistive divisor in 'positive', $+I_0$. If $a_i = -1$, then the standard electrical current is switched

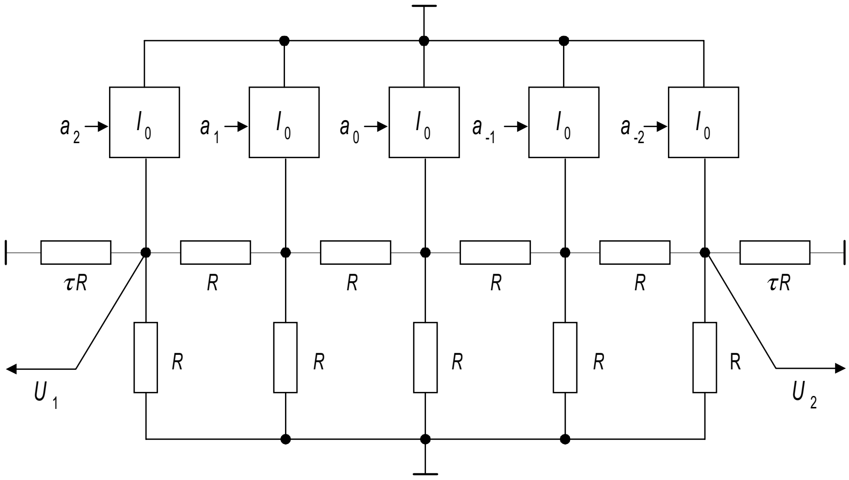

FIGURE 10. The golden ternary mirror-symmetrical DAC.

to the corresponding point of the golden mirror-symmetrical resistive divisor in 'negative', $-I_0$. Finally if $a_i = 0$, then the standard electrical current I_0 is not switched to the corresponding connection point.

The golden mirror-symmetrical DAC has the two mirror-symmetrical outputs, U_1 and U_2. One may demonstrate [12] that the outputs U_1 and U_2 are expressed, depending on the input of the ternary mirror-symmetrical code $a_m a_{m-1} \ldots a_0$, $a_{-1} a_{-2} \ldots a_{-m}$, as

$$U_1 = U_2 = \frac{I_0}{2} \sum_{i=-m}^{m} a_i \tau^{2i},$$

where a_i is the ith digit ternary numeral.

One of the most useful properties of the golden ternary mirror-symmetrical ADC in Figure 10 is an availability of two equal outputs of the ADC, U_1 and U_2. This property can be used for checking ADC since a breach of the equality $U_1 = U_2$ is an indication of DAC error.

7.2. Digital signal and analog-to-digital processors

A elegant solution of the ADC and DAC problem in Bergman's number system and in the ternary mirror-symmetrical number system allows the proposition that the field of digital signal processing and analog-to-digital processors can become the most effective applications of Bergman's number system and the ternary mirror-symmetrical number system considered in the present article. Precisely in this field, demands relating to speed of processor functioning in real time are very high. Since such processors are used in many real control systems then demands for information reliability are very high too. That is why the approach based on application of Bergman's number system and the ternary mirror-symmetrical number system might allow for development of fast reliable digital signal and analog-to-digital processors.

The following fact confirms this assumption. It is used very often in special signal transforms (Fourier's and other ones), which are the basic transforms performed by digital signal processors. A main problem of this computer direction is elaboration of fast discrete transforms. In this connection there is a great interest in the work of the Russian scientist Vladimir Chernov who is developing new discrete transforms based on Fibonacci numbers and golden ratio applications [14]. The abstract of Chernov and Pershina's article [14] claims: 'In the article the methods of an "error-free" computation of a discrete circular convolution via Mersenne and Fermat number-theoretical transforms (NTT) are considered. The essential extension of a range of length NTT is shown, computed without multiplication and is achieved by means of the representation of data in the Fibonacci number system'.

The Fibonacci number system mentioned in the abstract is similar to Bergman's number system. The article [14] claims that to get fast discrete transforms without multiplication it is necessary to represent initial data in Bergman's number system! There is no doubt that the ternary mirror-symmetrical number system considered above might become a source for new fast discrete transforms using data ternary mirror-symmetrical representation.

REFERENCES

[1] Malinovskiy, B. N. (1995) *History of Computers in the Faces*. 'KIT', Kiev (in Russian).

[2] Brousentsov, N. P. (1965) An experience of the ternary computer development. *Bulletin of Moscow University, Mathematics and Mechanics*, **2**, 39–48.

[3] Brousentsov, N. P., Maslov, S. P., Rosin, V. P. and Tishulina, A. M. (1965) *Small Digital Computing Machine 'Setun'*. Moscow State University, Moscow (in Russian).

[4] Brousentsov, N. P. and Zhogolev, E. A. (1971) The structure and algorithmic description of small computing machine. Collection of articles. *Computers and Problems of Cybernetics*, Vol. 8. Leningrad State University, Leningrad (in Russian).

[5] Brousentsov, N. P., Zhogolev, E. A. and Maslov, S. P. (1979) General characteristic of small digital machine 'Setun 70'. Collection of articles. *Computers and Problems of*

Cybernetics, Vol. 16. Leningrad State University, Leningrad (in Russian).

[6] Pospelov, D. A. (1970) *Arithmetical Foundations of Digital Computers*. Vyshaja Shkola, Moscow (in Russian).

[7] Knuth, D. E. (1997) *The Art of Computer Programming, vol. 2*. Semi-numerical Algorithms, 3rd edn. Addison-Wesley, MA.

[8] Bergman, G. (1957) A number system with an irrational base. *Math. Mag.*, **31**, 98–119.

[9] Hoggat, V. E. (1969) *Fibonacci and Lucas Numbers*. Houghton-Mifflin, Palo Alto, CA.

[10] Stakhov, A. P. (1984) *Codes of the Golden Proportion*. Radio Communication, Moscow (in Russian).

[11] Collection of article (1989) *Noise-tolerant codes: Fibonacci computers*. Znanie, Moscow (in Russian).

[12] Stakhov, A. P. (1997) *Computer Arithmetic based on Fibonacci Numbers and Golden Section: New Information and Arithmetic Computer Foundations*, SKILLSET Training, Toronto. http://www.goldenmuseum.zibys.com.

[13] Licomendes, P. and Newcomb, R. (1984) Multilevel Fibonacci conversion and addition. *The Fibonacci Quarterly*, **22,** 196–203.

[14] Chernov, V. M. and Pershina, M. V. (1999) Fibonacci–Mersenne and Fibonacci–Fermat discrete transforms. Collection of papers, 'The Golden Section: Theory and Applications'. *Bol. Informat.*, **9/10**, 25–31.