British Journal of Mathematics & Computer Science
17(1): 1-49, 2016, Article no.BJMCS.25969
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Fibonacci *p*-codes and Codes of the "Golden" *p*-proportions: New Informational and Arithmetical Foundations of Computer Science and Digital Metrology for Mission-Critical Applications

Alexey Stakhov[1*]

[1]International Club of the Golden Section, Canada.

Author's contribution

The sole author designed, analyzed and interpreted and prepared the manuscript.

Article Information

DOI: 10.9734/BJMCS/2016/25969
Editor(s):
(1) Wei-Shih Du, Department of Mathematics, National Kaohsiung Normal University, Taiwan.
Reviewers:
(1) Santiago Tello-Mijares, Instituto Tecnológico Superior de Lerdo, Mexico.
(2) Danyal Soybas, Erciyes University, Turkey.
(3) Manuel Alberto M. Ferreira, ISCTE-IUL, Portugal.
Complete Peer review History: http://sciencedomain.org/review-history/14921

Original Research Article

Received: 28th March 2016
Accepted: 2nd June 2016
Published: 7th June 2016

Abstract

In the 70s and 80s years of the past century, the new positional numeral systems, called Fibonacci *p*-codes and codes of the golden *p*-proportions, as new informational and arithmetical foundations of computer science and digital metrology, was considered as one of the most important directions of Soviet computer science and digital metrology for mission-critical applications. The main advantage of this direction was to improve the information reliability of computer and measuring systems. This direction has been patented abroad widely (more than 60 patents of US, Japan, England, France, Germany, Canada and other countries). Theoretical basis of this direction have been described in author's books "Introduction into the Algorithmic Measurement Theory" (1977), and "Codes of the Golden Proportion" (1984). Unfortunately, these books didn't be translated into English and, therefore, the Soviet scientific achievements in this field were virtually unknown for Western and world experts in computer science and digital metrology. Under author's leadership, a number of interesting engineering developments have been carried out. Some of them (self-correcting 18-bit analog-to-digital and digital-to- analog converters) exceeded the world level. Unfortunately, after the collapse of the Soviet Union in 1991, government funding of these developments was stopped. However, theoretical developments in this area continued. The purpose of this article is to state the history and the main scientific and engineering achievements in this field, as one of the important direction in the improvement of informational reliability of computer and measuring systems for mission-critical applications.

Corresponding author: E-mail: goldenmuseum@rogers.com;

Keywords: Fibonacci numbers; golden ratio; Pascal triangle; Fibonacci p-numbers; golden p-proportions; Fibonacci p-codes; Bergman system; codes of the golden p-proportions; Fibonacci arithmetic; "golden" arithmetic; ternary mirror-symmetrical arithmetic; "golden" resistive divisors; "golden" DAC and ADC; "golden" self-correcting ADC.

1 Instead Introduction: "Trojan Horse" of Computers for Mission-critical Applications

1.1 The main disadvantage of the binary system

As is known, the binary system was introduced into the computer science by **John von Neumann** (1903-1957) (together with von Neumann's colleagues at the Princeton Institute for Advanced Study) in 1946.

Fig. 1. John von Neumann
Hungarian-American pure and applied mathematician, physicist, inventor, computer scientist and polymath

A justification of the use of the binary system in electronic computers was one of the most important "*von Neumann's principles"* [1]. In that time, this proposal was absolutely right decision, because the binary system was best suited to binary nature of the electronic components and Boolean logic. Moreover, it should take into account the fact that at that time other alternative numeral systems in science simply did not exist. The choice was very small: *decimal, ternary* or *binary* systems. Preference was given to the binary system. However, along with the binary system its *"Trojan horse"* (according to the apt saying of Russian expert in Computer Science academician Hetagurov [2] was introduced into computer technology and digital metrology. We are talking on *zero redundancy* of the binary system. "Zero redundancy" means that all binary codeword's are "allowed," what makes impossible detecting any random errors and constant faults that inevitably (with more or less likely) may appear in electronic systems under the influence of various external and internal factors (radiation, electromagnetic feedback, tire noise power, etc.). Thus, the next not very optimistic conclusion follows from this consideration. **Mankind became hostage of the classical binary system**, which is the basis of modern information technology and digital metrology. Therefore, the further development of microprocessor and information technology, based on the binary system, should be declared inadmissible for certain areas of mission-critical applications. The binary system can not serve as the informational and arithmetical basis for specialized computer and measuring systems (space systems, fast transport, complex technological objects, nuclear systems and so on), as well as nano-electronic systems, where problems of reliability, noise immunity, stability, survivability of computer and measuring systems come to the fore.

Unfortunately, microelectronics was forced to adopt all technical solutions of the classical computer technology, along with the binary system. In this field, the "Trojan horse" of binary system (**zero redundancy**) has moved to microprocessors and microcontrollers.

Currently, the binary system together with its "Trojan horse" begins to take its firm positions in nano-electronics [3] what can lead to unpredictable consequences for the further development of information technology. Reducing the size of electronic components increases the probability of random errors and faults what is the reason for the reduction of noise immunity of electronic systems. **Therefore, a digital micro-electronics and nano-electronics in particular has become one of the mission-critical areas for the use of the binary system.**

1.2 The first author's publications on the redundant positional numeral systems

For the first time, the study on the redundant methods of positional representation of numbers was performed by the author in the early 70s of the 20th century in the Taganrog Radio Engineering Institute (Russia) (1971-1977) [4]. At the same time, the first author's articles [5-7] on the Fibonacci *p*-codes had been published. In 1978 author has introduced a concept of the codes of the golden *p*-proportions [6,7]. The fundamentals of the Fibonacci *p*-codes and codes of the golden *p*-proportions were analyzed in author's articles and books [8-15].

1.3 Fibonacci patenting

In 1976 the author worked as Visiting-Professor of the Vienna University of Technology (Austria). At the final stage of the stay in Austria, the author made the speech "Algorithmic measurement theory and foundations of computer arithmetic's" at the joint meeting of the Austrian Cybernetics and Computer Societies. The speech aroused great interest of Austrian scientists. In this connection, the Soviet Ambassador in Austria **Efremov** had written to the Soviet State Committee on Science and Technology the letter, which contained the following proposal:

> *"Taking into consideration the interest of the Austrian scientists in Prof. Stakhov invention in the field of new numeral system, based on Fibonacci numbers, it would be appropriate speeding up the process of patenting invention by Prof. Stakhov abroad what will preserve a priority of Soviet science in this computer field and possibly will give economic benefit. "*

The proposal of the USSR Ambassador in Austria was approved on the higher governmental and scientific levels of Soviet Union and, since 1976, the widespread patenting of author's inventions in the field of "Fibonacci computers" was launched in all the leading countries-producers of computer technology (USA, Japan. England, Germany, France, Canada and other countries).

The main purpose of patenting was to protect a priority of Soviet science in the field of "Fibonacci computers" and to make a technological leap in the mission-critical computer applications (in particular for space systems).

New computer arithmetic was the subject of patenting. But in accordance with the patent laws of most countries, it is impossible to patent mathematical achievement, in particular, computer arithmetic and codes based on Fibonacci numbers or "golden ratio".

That is why, it was decided protecting new computer arithmetic and codes, based on Fibonacci numbers and "golden ratio," by using specialized original device, which could implement this arithmetic and codes. With this purpose, the specialized operating device for Fibonacci *p*-codes and codes of the golden *p*-proportions was designed. It was *pioneering invention*, which allowed implementing all other operating computer and measuring devices (registers, counters, adders, analog-to-digit and digit-to-analog converters and so on). As a result of these reasoning's, there appeared the idea of multi-section invention formula, where the first section described the pioneering invention for Fibonacci and "golden" computing and measuring systems.

What are the results of this unprecedented patenting? Soviet inventions in the field of "Fibonacci's and "golden" computing and measuring systems" were protected with more than 60 patents of USA, Japan, England, France, Germany, Canada and other countries. This patenting showed that the idea of the "Fibonacci and "golden" computing and measuring systems" is a completely new and original, but these patents [16-28] are official legal documents confirming the priority of Soviet science (and the author of this article) in the creation of new direction in the field of computer technology and digital metrology.

In June 1989, on the initiative of the President of the Ukrainian Academy of Sciences **Boris Paton**, this direction was discussed at a special meeting of the Presidium of the Ukrainian Academy of Sciences.

1.4 The dramatic fate of Fibonacci's and "golden" developments in the Soviet Union

Unfortunately, the so-called "Gorbachev's perestroika" and the collapse of the Soviet Union in 1991, when Soviet Union broke up into a number of independent states (Russia, Ukraine, Belarus, etc.) caused irreparable blow on this scientific direction. Since 1989, governmental funding of this direction was sharply reduced and then completely stopped in 1991. But this does not mean that the conception of "Fibonacci computers" is obsolete. On the contrary, at the present stage of the development of computer technology and digital metrology, this conception has become even more relevant in the modern field of computer technology and digital metrology, especially in microprocessor and nano-electronic technology. This circumstance became the main motive to return again to these ideas [29-34], developed by the author in the 70s-80s of the 20th century (see author's books [14,15]).

In conclusion, it should be noted that along with the Soviet studies on "Fibonacci Arithmetic" and "Fibonacci computers", in the same period, such studies have been performed in the United States (University of Maryland) [35-40].

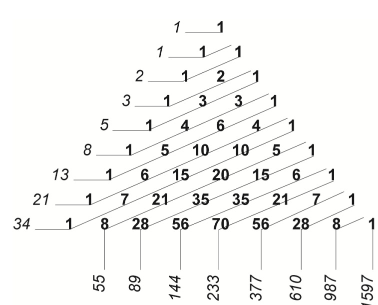
Studies of the American [35-40], Soviet [4-28] and also Ukrainian and Canadian [29-34] scientists in this field are confirmation of the fact that, since 70s years of the 20th century, the terms "Fibonacci code," "Fibonacci arithmetic" and "Fibonacci computer" became widely known both in the American, Soviet, Ukrainian and Canadian scientific and technical literature.

2 Fibonacci *p*-codes and Their Peculiarities

2.1 Pascal triangle and Fibonacci numbers

In the book [41], the famous American mathematician and populariser of mathematics **George Polya** (1887 - 1985) has found a surprising connection between *Fibonacci numbers* and *"diagonal sums" of Pascal's triangle* (see Table 1)

Table 1. Diagonal sums of Pascal's triangle



If we calculate the sums of binomial coefficients standing on the diagonals:

1=1, 1=1, 2=1+1, 3=1+2, 5=1+3+1, 8=1+4+3, 13=1+5+6+1, 21=1+6+10+1, …, we get the Fibonacci sequence, that is, Pascal's triangle is a "generator" of the Fibonacci numbers!

At first glance, it seems that finding this connection is so simple and so "elementary" what it hardly worthy the attention of mathematicians. However, this mathematical result, which, as say, "lay on the surface," for several centuries was the "big secret" for **Blaise Pascal** (1623–1662) and other mathematicians who studied *Fibonacci numbers* and *Pascal's triangle*. However, the surprisingly simple mathematical connection between Fibonacci numbers and Pascal triangle opens the way to the deep union of two important mathematical theories, the *theory of Fibonacci numbers* [42-44] and *combinatorics* and this union can become a fruitful source for new mathematical ideas and generalizations.

2.2 Fibonacci *p*-numbers

The development of *Polya's idea* [41] led the author [14] to the discovery of the surprising generalized recurrence relation, which "generates" an infinite number of the new recurrence sequences, *Fibonacci p-numbers* (*p*=0,1,2,3,…) given by the following recurrence relation:

$$F_p(i) = F_p(i-1) + F_p(i-p-1) \text{ for } i > p+1 \tag{1}$$

at the following initial terms:

$$F_p(1) = F_p(2) = \ldots = F_p(p+1) = 1. \tag{2}$$

Here the numbers *p*=0,1,2,3,… correspond to the different inclinations of diagonals in Pascal's Triangle (see Table 1).

Note that the recurrence relation (1) at the seeds (2) generates many remarkable numerical sequences, in particular, the *binary sequence* for the case *p*=0:

$$1,2,4,8,16,32,64,\ldots,2^{n-1},\ldots, \tag{3}$$

and the *Fibonacci sequence* for the case *p*=1:

$$1,1,2,3,5,8,13,21,34,55,\ldots, F_n,\ldots \tag{4}$$

2.3 The "golden" *p*-proportions

A study of the limit of the ratios of two adjacent *Fibonacci p-numbers* [14,31] led to the following algebraic equation:

$$x^{p+1} - x^p - 1 = 0, \tag{5}$$

which is a generalisation of the "golden" algebraic equation $x^2 - x - 1 = 0$ with the positive root $\Phi = \frac{1+\sqrt{5}}{2}$ (the golden ratio).

In general, the positive roots Φ_p of the equation (5) are new mathematical constants called the *golden p-proportions* (*p*=0,1,2,3,…) that are a generalization of the classical golden proportion $\Phi = \frac{1+\sqrt{5}}{2}$.

Table 2. Partial cases of Φ_p

p	0	1	2	3	4
Φ_p	2	1.618	1.4656	1.3802	1.3247

Thus, we have every right to claim that *Pascal's triangle* is a "universal generator" of new recurrent sequences $F_p(n)$, which are a generalization of the classical Fibonacci numbers 1,1,2,3,5,8,13,… and new mathematical constants Φ_p (see Table 2), which are a generalization of the classical golden ratio. These mathematical concepts underlie the "*theory of the Fibonacci p-numbers and golden p-proportions,"* [4-15], which is a generalization of the classical *"theory of Fibonacci numbers and golden ratio*" [41-43].

2.4 Definition of the Fibonacci *p*-codes

It is proved in [14] that for the given integer $p \in \{0,1,2,3,...\}$, any natural number *N* can be represented as the following sum:

$$N = a_n F_p(n) + a_{n-1} F_p(n-1) + ... + a_i F_p(i) + ... + a_1 F_p(1), \tag{6}$$

where $a_i \in \{0,1\}$ is a bit of the i-th digit of the positional representation (6); n is a number of bits of the code (6).; $F_p(i)(i = 1,2,3,...,n)$ is the *Fibonacci p-number,* the weight of the i-th digit of the positional representation (6).

The sum (6) is called *Fibonacci p-code* [14].

In the Fibonacci *p*-code (6), the weights of digits are linked with the recurrence relation (1), which "generates" *Fibonacci p-numbers* starting from the seeds (2).

Abridged notation of the sum (6) looks as follows:

$$N = a_n a_{n-1} ... a_i ... a_1 . \tag{7}$$

The abridged notation (7) is called *Fibonacci representation* of natural number *N*.

2.5 Partial cases of the Fibonacci *p*-codes

Note that the sum (6) includes an infinite number of different binary (0,1) positional representations, because every $p(p = 0,1,2,3,...)$ "generates" its own positional representation of the kind (6).

Let $p = 0$. For this case, Fibonacci (*p*=0)-numbers $F_0(i)$ coincide with the "binary" numbers, ie, $F_0(i) = 2^{i-1}$ and therefore the sum (6) takes the form of the classical binary code for natural numbers:

$$N = a_n 2^{n-1} + a_{n-1} 2^{n-2} + ... + a_i 2^{i-1} + ... + a_1 2^0 . \tag{8}$$

Let $p=1$. For this case, Fibonacci (p=1)-numbers $F_1(i)$ coincide with the classical Fibonacci numbers, ie, $F_1(i)=F_i$ and for this case the sum (6) takes the following form:

$$N=a_nF_n+a_{n-1}F_{n-1}+...+a_iF_i+...+a_1F_1. \quad (9)$$

Recall that the weight of i-th digits F_i in the classical Fibonacci code (9) are linked by the classical Fibonacci recurrence relation:

$$F_i=F_{i-1}+F_{i-2}; \quad F_1=F_2=1. \quad (10)$$

Let now $p=\infty$. For this case all Fibonacci ($p=\infty$)-numbers, given by (1), (2) are equal to 1 identically, ie, for any i we have: $F_p(i)=1$. For this case, the sum (6) takes the following form, called *"unitary code"*:

$$N=\underbrace{1+1+...+1}_{N}. \quad (11)$$

Thus, the Fibonacci p-codes (6) is a wider generalization of the "binary code" (8) (the case $p=0$). The classical Fibonacci code (9) ($p=1$) and "unitary code" (11) $(p=\infty)$ are partial cases of the Fibonacci p-codes (6).

The next important conclusions follow from this consideration:

1) The first conclusion is the fact that the Fibonacci p-codes (6) are a generalization of the "unitary code" (11) $(p=\infty)$. But, by its form the "unitary code" (11) coincides with Euclidean definition of natural numbers, which gives beginning of *number theory,* one of fundamental theories of mathematics. We can hypothesize from this fact that the Fibonacci p-codes (6) can be viewed as a new definition of natural numbers, which implies the idea of new number theory, based on Fibonacci p-numbers.
2) The second conclusion is the fact that the Fibonacci p-codes (6) are generalization of the "binary system" (8) ($p=0$). But we should not forget that the binary system (8) is the basis of modern computers! But then we came to the following idea. Because the Fibonacci p-codes (6) are new method of the binary (0,1) positional representation of numbers, this means that we can come to a new class of computers, *Fibonacci computers* as a new direction in the development of computer technology!
3) Because the *Fibonacci p-codes* (6) are based on *the Fibonacci p-numbers*, which follow from the *Pascal Triangle*, this means that the *Fibonacci p-codes* (6) are new unique mathematical object, which unites three fundamental conceptions of modern science: *number theory*, which is the basis of mathematics, *combinatorics,* based on the *Pascal Triangle*, and finally, *computer science*, based on the *binary system* (8).

3 Properties of Fibonacci Representations

3.1 A range of number representation in binary code

Consider the set of the n-digit binary words. A number of them is equal to 2^n. For the classical binary code (8) (p = 0) the mapping of the n-digit binary words onto the set of natural numbers has the following peculiarities:

a) *Uniqueness of mapping.* This means that for the infinite n there is one-to-one correspondence between natural numbers and sum (7), that is, each integer N has the only representation in the form (7).
b) For the given n, by using the binary code (8) we can represent all integers in the range from 0 to $2^n - 1$, that is, the range of number representation is equal to 2^n.
c) The minimal number 0 and the maximal number $2^n - 1$ have the following binary representations in the binary code (8), respectively:

$$0 = 0\,0 \ldots 0$$
$$2^n - 1 = 1\,1 \ldots 1$$

3.2 A range of number representation in the Fibonacci *p*-codes

For the Fibonacci p-code (6) the mapping of the n-digit binary words onto natural numbers has distinct peculiarities for the case $p > 0$.

Let n =5. Then for the cases p =1 and p =2 the mappings of the 5-digit Fibonacci p-code (6) onto the natural numbers have the form, represented in Tables 3 and 4, respectively.

Table 3. Mapping of the Fibonacci 1-code onto natural numbers

CC	5	3	2	1	1	N	CC	5	3	2	1	1	N
A_0	0	0	0	0	0	0	A_{16}	1	0	0	0	0	5
A_1	0	0	0	0	1	1	A_{17}	1	0	0	0	1	6
A_2	0	0	0	1	0	1	A_{18}	1	0	0	1	0	6
A_3	0	0	0	1	1	2	A_{19}	1	0	0	1	1	7
A_4	0	0	1	0	0	2	A_{20}	1	0	1	0	0	7
A_5	0	0	1	0	1	3	A_{21}	1	0	1	0	1	8
A_6	0	0	1	1	0	3	A_{22}	1	0	1	1	0	8
A_7	0	0	1	1	1	4	A_{23}	1	0	1	1	1	9
A_8	0	1	0	0	0	3	A_{24}	1	1	0	0	0	8
A_9	0	1	0	0	1	4	A_{25}	1	1	0	0	1	9
A_{10}	0	1	0	1	0	4	A_{26}	1	1	0	1	0	9
A_{11}	0	1	0	1	1	5	A_{27}	1	1	0	1	1	10
A_{12}	0	1	1	0	0	5	A_{28}	1	1	1	0	0	10
A_{13}	0	1	1	0	1	6	A_{29}	1	1	1	0	1	11
A_{14}	0	1	1	1	0	6	A_{30}	1	1	1	1	0	11
A_{15}	0	1	1	1	1	7	A_{31}	1	1	1	1	1	12

The analysis of Tables 3 and 4 allows finding the following peculiarities of the binary representations of natural numbers in the Fibonacci p-codes (6). By using the 5-digit Fibonacci 1-code (Table 2) we can represent 13 integers in the range from 0 to 12, inclusively. Note that the number 13 is the Fibonacci 1-number with the index 7, i.e. $F_1(7) = F_7 = 13$. We can see from Table 3 that by using the 5-digit Fibonacci 2-code we can represent 9 integers in the range from 0 to 8, inclusively, at that the number 9 is the Fibonacci 2-number with the index 8, i.e. $F_2(8) = 9$. The results of this consideration are partial cases of the following theorem [31].

Theorem 1. *For the given integers $n \geq 0$ and $p \geq 0$ by using the n-digit Fibonacci p-code we can represent $F_p(n+p+1)$ integers in the range from the minimal number 0 to the maximal number $F_p(n+p+1) - 1$, inclusively.*

Note that for the case p=0 $F_0(n+1) = 2^n$ and Theorem 1 is reduced to the well-known theorem about the number representation range, equal to 2^n for the n-digit binary code (8) .

Table 4. Mapping of the Fibonacci 2-code onto natural numbers

CC	3	2	1	1	1	N	CC	3	2	1	1	1	N
A_0	0	0	0	0	0	0	A_{16}	1	0	0	0	0	3
A_1	0	0	0	0	1	1	A_{17}	1	0	0	0	1	4
A_2	0	0	0	1	0	1	A_{18}	1	0	0	1	0	4
A_3	0	0	0	1	1	2	A_{19}	1	0	0	1	1	5
A_4	0	0	1	0	0	1	A_{20}	1	0	1	0	0	4
A_5	0	0	1	0	1	2	A_{21}	1	0	1	0	1	5
A_6	0	0	1	1	0	2	A_{22}	1	0	1	1	0	5
A_7	0	0	1	1	1	3	A_{23}	1	0	1	1	1	6
A_8	0	1	0	0	0	2	A_{24}	1	1	0	0	0	5
A_9	0	1	0	0	1	3	A_{25}	1	1	0	0	1	6
A_{10}	0	1	0	1	0	3	A_{26}	1	1	0	1	0	6
A_{11}	0	1	0	1	1	4	A_{27}	1	1	0	1	1	7
A_{12}	0	1	1	0	0	3	A_{28}	1	1	1	0	0	6
A_{13}	0	1	1	0	1	4	A_{29}	1	1	1	0	1	7
A_{14}	0	1	1	1	0	4	A_{30}	1	1	1	1	0	7
A_{15}	0	1	1	1	1	5	A_{31}	1	1	1	1	1	8

3.3 Plurality of number representation

Plurality of number representation in the form (6) is the next peculiarity of the Fibonacci *p*-codes (6) for the cases $p > 0$. By excepting the minimal number 0 and the maximal number $F_p(n+p+1) - 1$, the rest integers from the range $[0, F_p(n+p+1) - 1]$ have more than one representation in the form (6). This means that all integers in the range $[1, F_p(n+p+1) - 2]$ have multiple representations in the Fibonacci *p*-codes (6) for the cases $p > 0$.

Consider now the mapping of integers onto the 5-digit binary code combinations *A* in accordance with Tables 3 ($p = 1$) and Table 4 ($p = 2$).

Table 5. Mapping of natural numbers on binary code words for the Fibonacci 1- and 2-codes

$p = 1$	$p = 2$
$0 = \{A_0\}$	$0 = \{A_0\}$
$1 = \{A_1, A_2\}$	$1 = \{A_1, A_2, A_4\}$
$2 = \{A_3, A_4\}$	$2 = \{A_3, A_5, A_6, A_8\}$
$3 = \{A_5, A_6, A_8\}$	$3 = \{A_7, A_9, A_{10}, A_{12}, A_{16}\}$
$4 = \{A_7, A_9, A_{10}\}$	$4 = \{A_{11}, A_{13}, A_{14}, A_{17}, A_{18}, A_{20}\}$
$5 = \{A_{11}, A_{12}, A_{16}\}$	$5 = \{A_{15}, A_{19}, A_{21}, A_{22}, A_{24}\}$
$6 = \{A_{13}, A_{14}, A_{17}, A_{18}\}$	$6 = \{A_{23}, A_{25}, A_{26}, A_{28}\}$
$7 = \{A_{15}, A_{19}, A_{20}\}$	$7 = \{A_{27}, A_{29}, A_{30}\}$
$8 = \{A_{21}, A_{22}, A_{24}\}$	$8 = \{A_{31}\}$
$9 = \{A_{23}, A_{25}, A_{26}\}$	
$10 = \{A_{27}, A_{28}\}$	
$11 = \{A_{29}, A_{30}\}$	
$12 = \{A_{31}\}$	

Note that for the arbitrary *p* the minimal and the maximal numbers have the only binary representations in the *n*-digit Fibonacci *p*-codes:

$$N_{min} = 0 = \underbrace{00...0}_{n} \tag{12}$$

$$N_{\max} = F_p(n+p) - 1 = \underbrace{11...1}_{n} \quad (13)$$

3.4 "Convolution" and "devolution" of the Fibonacci digits

The different Fibonacci representations (7) of one and the same integer N in the Fibonacci p-codes (6) for the cases $p>0$ may be obtained each from other by means of the peculiar code transformations called "*convolution*" and "*devolution*" of the binary digits. These code transformations are performed in scope of one Fibonacci representations (7) and follow from the basic recurrence relation (1), which connects the adjacent digit weights of the Fibonacci p-code (6). The general idea of these code transformations is described in [31].

It is most simply to demonstrate the "convolution" and "devolution" of the Fibonacci representation (7) for the case p=1. In this case these operations are performed over the three adjacent digits, namely over the l^{th} , $(l\text{-}1)^{\text{th}}$ and $(l\text{-}2)^{\text{th}}$ digits. Consider the fulfillment of these operations for the Fibonacci 1-code (classical Fibonacci code (9)):

(a) *Convolution*

$$7 = \begin{cases} \mathbf{0} & \mathbf{1} & \mathbf{1} & 1 & 1 \\ 1 & 0 & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ 1 & 0 & 1 & 0 & 0 \end{cases} \quad (14)$$

(b) *Devolution*

$$5 = \begin{cases} \mathbf{1} & \mathbf{0} & \mathbf{0} & 0 & 0 \\ 0 & 1 & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ 0 & 1 & 0 & 1 & 1 \end{cases} \quad (15)$$

A procedure, which consists in the fulfillment of all possible "convolutions" or "devolutions" in the Fibonacci representation (7), is named a "*code convolution*" or "*code devolution*", respectively. It is easy to prove that the fulfillment of the "code convolution" or "code devolution" results in the so-called "*convolute*" or "*devolute*" Fibonacci representation of natural number N.

For the case p=1, the *"convolute"* and *"devolute"* Fibonacci representations of the number N have peculiar indications. In particular, *in the "convolute" Fibonacci representation two bits of 1 together do not meet* and *in the "devolute" Fibonacci representation two bits of 0 together do not meet, starting from the highest 1 of the Fibonacci representation (7).*

A rule of the "convolution-devolution inversion" is of great importance for technical applications. This rule consists of the following: *the "convolution" of the initial Fibonacci representation is equivalent to the "devolution" of the inverse Fibonacci representation and conversely.* By using this rule, we can fulfill the reduction to the "convolute" form for the example (14) as follows:

(a) Inversion of the initial Fibonacci representation 0 1 1 1 1:

1 0 0 0 0

(b) "Code devolution" of the inverse Fibonacci representation:

0 1 0 1 1

(c) Inversion of the obtained Fibonacci representation:

1 0 1 0 0 (the "convolute" form of the Fibonacci representation 0 1 1 1 1).

Consider now peculiarities of the "convolution" and "devolution" for the lowest digits of the Fibonacci 1-code. As is well-known in the case $p = 1$ the weights of the two lowest digits of the Fibonacci 1-code is equal to 1 identically, that is, $F_1 = F_2 = 1$. And then the operations of the "devolution" and "convolution" for these digits are performed as follows:

1 0 = 0 1 ("devolution") and 0 1 = 1 0 ("convolution").

3.5 The radix of the Fibonacci *p*-code

For the case $p = 0$ the radix of the binary system (8) is calculated as the ratio of the adjacent digit weights, that is,

$$\frac{2^k}{2^{k-1}} = 2.$$

Apply this principle to the Fibonacci p-code (6) and consider the ratio

$$\frac{F_p(k)}{F_p(k-1)} . \tag{16}$$

A limit of the ratio (16) is named *a radix of the Fibonacci p-code* (6). It is easy to prove that

$$\lim_{k\to\infty} \frac{F_p(k)}{F_p(k-1)} = \Phi_p , \tag{17}$$

where Φ_p is the golden p-proportion.

This means that the radix of the Fibonacci p-code (6) for the case $p>0$ is irrational number Φ_p .

3.6 A minimal form of the Fibonacci *p*-code

The following theorem is of great importance for a theory of the Fibonacci p-codes [31].

Theorem 2. *For the given integers $p\geq 0$ and $n\geq p+1$ the arbitrary integer N can be represented in the following unique form:*

$$N = F_p(n) + R_1, \tag{18}$$

where

$$0 \leq R_1 < F_p(n-p). \tag{19}$$

Note that for the case $p=0$ we have:

$$F_0(n) = 2^{n-1}$$

and then the expressions (18) and (19) take the following well-known form (for the "binary" arithmetic):

$$N = 2^{n-1} + R_1, \qquad 0 \leq R_1 < 2^{n-1}. \tag{20}$$

If we represent the integer N according to the formula (18) and then represent all remainders $R_1, R_2, ..., R_k$, arising as result of this representation, according to the same formula (18) up to obtaining the remainder equal to 0, we get a peculiar representation of the integer N in the Fibonacci p-code (6). Its peculiarity consists of the fact that in the Fibonacci representation of the integer N given by (7) no less than p bits of 0 follow after every bit $a_l = 1$ from the left to the right, that is,

$$a_{l-1} = a_{l-2} = ... = a_{l-p} = 0.$$

Such Fibonacci representation of the integer N is called **MINIMAL FORM** or *minimal Fibonacci representation of the integer N in the Fibonacci p-code* (6). This name reflects the fact that for the case $p = 1$ the **MINIMAL FORM** of the integer N has a minimal number of the bits of 1 in the Fibonacci representation of the Fibonacci 1-code (9) among all Fibonacci representations of the same integer N.

For example, by using the above algorithm, we can obtain the following **MINIMAL FORMS** of the number 25 for the Fibonacci 1- and 2-codes (Table 6).

Table 6. MINIMAL FORMS for the Fibonacci 1- and 2-codes

	$F_1(i)$	**55**	**34**	**21**	**13**	**8**	**5**	**3**	**2**	**1**	**1**
$p = 1$	25 =	0	0	**1**	**0**	0	0	**1**	**0**	**1**	**0**
	$F_2(i)$	19	13	9	6	4	3	2	1	1	1
$p = 2$	25 =	**1**	**0**	**0**	**1**	**0**	**0**	0	0	0	0

A peculiarity of the Fibonacci representations of the number 25, given by Table 6, consists in the following. For the case p=1 not less then one bit of 0 follows after every bit of 1 from the left to the right in the Fibonacci representation of the number 25; for the case p=2 not less then two bits of 0 follow after every bit of 1 from the left to the right in the Fibonacci representation of the same number 25.

Corollary from Theorem 2. *For a given p (p=0, 1, 2, 3, ...) every integer N has the only minimal form in the Fibonacci p-code.*

This means that there is one-to-one mapping of natural numbers onto the minimal forms of the Fibonacci p-code (6).

Note that for the case p=0 (the classical binary code) every integer N has the only binary representation in the form (7). This means that every binary representation (7) is its "minimal form".

The following theorem is proved in [31].

Theorem 3. *For a given integer $p \geq 0$ in the minimal form of the n-digit Fibonacci p-code we can represent $F_p(n+1)$ integers in the range from 0 to $F_p(n+1)-1$, inclusively.*

For the case p=1 the MINIMAL FORM of the Fibonacci (p=1)-code has very simple indication: in the MINIMAL FORM two bits of 1 together do not meet. But the "convolute" Fibonacci representation, considered above, has the same property. This means that for the Fibonacci (p=1)-code the "convolute" form coincides with the MINIMAL FORM and the reduction of the Fibonacci representation for the Fibonacci (p=1)-code (9) to its MINIMAL FORM can be performed by using "convolutions". The example (14) demonstrates a process of the reduction of the Fibonacci representation for the Fibonacci (p=1)-code (9) to the MINIMAL FORM. Also the notion of the MAXIMAL FORM is very important for the Fibonacci (p=1)-code. The MAXIMAL FORM can be obtained from the initial Fibonacci representation (7) by using "devolutions" and the MAXIMAL FORM coincides with the "devolute" form. The example (15) demonstrates a process of the reduction of the minimal Fibonacci representation 10000 to the maximal Fibonacci representation 01011. Note that the operations of the reduction of the Fibonacci representations to their MINIMAL or MAXIMAL FORMS are the most important operations of the Fibonacci arithmetic.

3.7 Code redundancy of the Fibonacci *p*-codes

For the case p=0 the Fibonacci (p=0)-code (classical binary code) is non-redundant. But for the case p>0 all Fibonacci p-codes are redundant. And their redundancy shows itself in *plurality* of the Fibonacci representations of one and the same integer N. Theorems 1 and 2 allow calculating the code redundancy of the Fibonacci p-codes for the cases p>0 in comparison with the classical binary code (p = 0).

We can calculate the relative code redundancy r by the following formula [45]:

$$r = \frac{n-m}{m} = \frac{n}{m} - 1, \tag{21}$$

where n and m are the code length of the redundant and non-redundant codes, respectively. Note that the code redundancy definition, given by (21), characterizes a relative increasing of the code length of the redundant code in the comparison to the non-redundant code for the representation of one and the same number range.

Theorems 1 and 2 determines the ranges of number representations in Fibonacci p-codes for two cases: (*a*) when we use all possible Fibonacci representations (Theorem 1) and (*b*) when we use only MINIMAL FORMS of Fibonacci representations (Theorem 2). For the case (*a*) we can represent $F_p(n+p+1)$ integers, for the case (*b*) we can represent $F_p(n+1)$ integers

For the code representation of numbers in the number range $F_p(n+p+1)$ or $F_p(n+1)$, it is necessary to use either $m_1 \approx \log_2 F_p(n+p+1)$ or $m_2 \approx \log_2 F_p(n+1)$ binary digits of the non-redundant code, respectively. Using (21) we can obtain the following formulas for the calculation of the relative code redundancy of the Fibonacci p-code (for the case p=1 and p=2, respectively):

$$r_1 = \frac{n}{\log_2 F_p(n+p+1)} - 1 \ ; \tag{22}$$

$$r_2 = \frac{n}{\log_2 F_p(n+p+1)} - 1 . \tag{23}$$

The simplest redundant Fibonacci p-code is the code, corresponding to the case p=1. We can calculate the limiting value of the relative redundancy for this code. For this case the formulas (22), (23) take the following forms, respectively:

$$r_1 = \frac{n}{\log_2 F_{n+2}} - 1 \ ; \tag{24}$$

$$r_2 = \frac{n}{\log_2 F_{n+1}} - 1 , \tag{25}$$

where F_{n+2}, F_{n+1} are the classical Fibonacci numbers.

We can represent the Fibonacci numbers F_{n+2} and F_{n+1} by using Binet formulas [31]:

$$F_n = \begin{cases} \dfrac{\Phi^n + \Phi^{-n}}{\sqrt{5}} & for \quad n = 2k+1; \\ \dfrac{\Phi^n - \Phi^{-n}}{\sqrt{5}} & for \quad n = 2k \end{cases} \tag{26}$$

For a large n we can write Binet formulas (26) in the following approximate form:

$$F_n \approx \frac{\Phi^n}{\sqrt{5}} \tag{27}$$

Using (27) and substituting the approximate values for F_{n+2} and F_{n+1} into the formulas (24) and (25), we can obtain the following formulas:

$$r_1 = \frac{n}{(n+2)\log_2\Phi - \log_2\sqrt{5}} - 1\,; \tag{28}$$

$$r_2 = \frac{n}{(n+1)\log_2\Phi - \log_2\sqrt{5}} - 1\,. \tag{29}$$

If we aim $n\to\infty$ in the expressions (28) and (29), we can see that they coincide for the case $n\to\infty$. Here the limiting value of the relative code redundancy for the Fibonacci 1-code (9) is determined by the following expression:

$$r = \frac{1}{\log_2\Phi} - 1 = 0.44\,. \tag{30}$$

Thus, the limiting value of the relative code redundancy of the Fibonacci 1-code is a constant value equal to 0.44 (44%).

4 Fibonacci Arithmetic

4.1 Comparison of numbers in the Fibonacci *p*-codes

As it is shown above, Fibonacci *p*-codes (6) are a new class of positional numeral systems. Fibonacci *p*-codes (6) are similar to the binary system and are its generalization. Therefore, all the well-known properties of the positional numeral systems, in particular, the binary system, can be used to create the Fibonacci arithmetic, although the difference consists of the fact that we should take into consideration fundamental features of the Fibonacci *p*-codes, in particular, such as the *plurality* of number representation and *MINIMAL FORM*.

Let us begin from the comparison of numbers in the Fibonacci *p*-codes. A comparison of Fibonacci representations A and B is carried out in the Fibonacci *p*-codes similarly to the classical binary code, if before the comparison we reduce the compared Fibonacci representations to the MINIMAL FORM. This property (simplicity of number comparison) is one of the important arithmetical advantages of the Fibonacci *p*-codes.

For example we need to compare two Fibonacci representations A = 00111101101 and B = 00111110110, represented in the Fibonacci (p=1)-code (9) (the classical Fibonacci code). The comparison of numbers is performed in two steps:

1. Reduction of the compared Fibonacci representations to the MINIMAL FORM:

   A = 01010010010 and B = 01010100000. (31)

2. Digit-by-digit comparison of the MINIMAL FORMS (31), starting since the highest digit until obtaining first pair of the distinct bits:

   A = 01010**0**10010;
   B = 01010**1**00000.

We can see that the first distinct pair of bits for the compared Fibonacci representations *A* and *B* contains the bit of 0 in the MINIMAL FORM of the first Fibonacci representation *A* and the bit of 1 in the MINIMAL FORM of the second Fibonacci representation *B*. This means that *B*>*A*.

4.2 Fibonacci counters

Algorithm of the Fibonacci summing counter is based on the following rule:

1) Before adding of bit 1 to LSB, the initial Fibonacci representation, corresponding to the number *N*, is reduced to such a form that the value of LSB will be equal to 0.
2) After that, we add the bit 1 to LSB what leads to increasing of number in the Fibonacci counter to the value *N*+1.
3) We repeat point 1 and 2 until overflowing Fibonacci counter.

Let us demonstrate this algorithm in the following **Example 1.**

Example 1. Algorithm of the Fibonacci summing counter

$$\begin{aligned}
&000000+1=0000\boxed{\mathbf{01}}=000010=1\\
&000010+1=000\boxed{\mathbf{011}}=000100=2\\
&000100+1=0001\boxed{\mathbf{01}}=000110=3\\
&00\boxed{\mathbf{011}}0+1=0010\boxed{\mathbf{01}}=001010=4\\
&001010+1=001\boxed{\mathbf{011}}=001100=5\\
&0\boxed{\mathbf{011}}00+1=0100\boxed{\mathbf{01}}=010010=6\\
&010010+1=010\boxed{\mathbf{011}}=010100=7\\
&010100+1=0101\boxed{\mathbf{01}}=010110=8\\
&01\boxed{\mathbf{011}}0+1=\boxed{\mathbf{011}}0\boxed{\mathbf{01}}=100010=9\\
&100010+1=100\boxed{\mathbf{011}}=100100=10\\
&100100+1=1001\boxed{\mathbf{01}}=100110=11\\
&10\boxed{\mathbf{011}}0+1=1010\boxed{\mathbf{01}}=101010=12\\
&101010+1=101\boxed{\mathbf{011}}=1\boxed{\mathbf{011}}00=\boxed{\mathbf{110}}000=000000
\end{aligned} \tag{32}$$

Here we single out in bold in parentheses those situations where we can perform “convolutions” in Fibonacci representations. Consider, for example, the situation of the transition of the Fibonacci representation of the number 8 to the Fibonacci representation of the number 9. In this case, while recording of it 1 to LSB of Fibonacci representation, we carry out convolution of bits 1 from the 2nd and 3rd digits in the 4th digit of Fibonacci representation ($011 \to 100$) what results to the Fibonacci representation of the number 9 ($\boxed{\mathbf{011}}00\boxed{\mathbf{01}} \to \mathbf{1000}0\mathbf{10=9}$).Note that the bottom row of the table (32) corresponds to the overflow of Fibonacci counter.

Subtraction of 1's in the Fibonacci code (9) is carried out by subtracting of bit 1 from the LSB of the Fibonacci representation of the number N, where the value of the LSB is equal to 1. Consider the **Example 2** of the functioning of the Fibonacci subtracting counter.

Example 2. Algorithm of the Fibonacci subtracting counter

$$\begin{aligned}
&1111-1=11\boxed{\mathbf{10}}=1101=6\\
&1101-1=1\boxed{\mathbf{100}}=1011=5\\
&1011-1=10\boxed{\mathbf{10}}=1001=4\\
&\boxed{\mathbf{100}}1-1=0110=0101=3\\
&0101-1=0\boxed{\mathbf{100}}=0011=2\\
&0011-1=00\boxed{\mathbf{10}}=0001=1\\
&0001-1=0000=0
\end{aligned} \tag{33}$$

Here we single out in bold in parentheses those situations, when we can realize "devolutions" in Fibonacci representations. Thus, a feature of the Fibonacci subtracting counter consist of the fact that in any situation the transition from the Fibonacci representation of the number N to the Fibonacci representation of the number N-1 is performed during sequential operations of the "devolutions."

The above algorithms of summing and subtracting Fibonacci counters show that these counters the prerequisites for the construction of high-speed Fibonacci counters (without the use of complex schemes of group transfer). That is, these simple examples shows certain advantages of the Fibonacci code (9) in comparison to the classical binary code (8).

It is necessary to pay attention to the latest development in the field of Fibonacci counters [32,33]. A peculiarity of new Fibonacci counter, described in [32,33], lies in the fact that here we use only minimal Fibonacci representations what improves noise-immunity of Fibonacci counter.

4.3 Fibonacci summation and subtraction

It is well known, the classical binary summation is based on the following elementary identity for the binary numbers:

$$2^k+2^k=2^{k+1} \tag{34}$$

where $2^k, 2^{k+1}$ are the weights of the k-th and (k+1)-th digits of the binary code (8), respectively .

The "deduction" of the Fibonacci p-summation rule begins from the analysis of the sum:

$$F_p(k)+F_p(k) \tag{35}$$

where $F_p(k)$ is the weight of the k-th digit of the Fibonacci p-code (6).

Let p = 1. For this case we have

$$F_p(k) = F_k , \tag{36}$$

where F_k is the classical Fibonacci numbers: 1,1,2,3,5,8,13,…, given by the recurrence relation (10). By using (10), we can represent the sum (35) as follows:

(a) $F_k + F_k = F_k + F_{k\text{-}1} + F_{k\text{-}2}$ (37)

(b) $F_k + F_k = F_{k+1} + F_{k\text{-}2}$ (38)

It follows from (37), (38) the following table for the Fibonacci summation (p=1) of the two binary digits $a_k + b_k$ with the same index k.

Table 7. Fibonacci summation for the case *p*=1

0	+	0	=	0	
0	+	1	=	1	
1	+	0	=	1	
1	+	1	=	1 1 1	(a)
1	+	1	=	1 0 0 1	(b)

It follows from Table 7 that the rules of the Fibonacci summation (p=1) coincide with the binary summation for the cases: 0+0=0, 0+1=1, 1+0=1; but for the case 1+1, the rules of the Fibonacci summation (p=1) don't coincide with the binary summation. For the case 1+1, the rules of the Fibonacci summation (p=1) are reduced to the following:

Rule 1. At the summation of the binary 1's of the k-th digit of the summand Fibonacci representations, the carry-out of two bits of 1 from the k-th digit to the other two digits arises.

Rule 2. There are two methods of carry-over formation. For the method (a) the bit of 1 is written to the k-th digit of the intermediate sum and the two carry-over's of 1 arise to the next two lower digits, namely to the (k-1)-th and (k-2)-th digits.

The method (b) assumes another rule of the Fibonacci summation (p=1) of the summand Fibonacci representations. The bit 0 is written to the k-th digit of the intermediate sum and the two carry-over's of 1 arise to the other digits, namely to the (k+1)-th and (k-2)-th digits.

The summation of the multi-digit numbers in the Fibonacci code (p=1) is fulfilled in accordance with the Table 7. However, we should follow the following rules:

Rule 3. Before the summation the summand Fibonacci representations are reduced to the MINIMAL FORM.

Rule 4. In accordance with Table 7 it is necessary to form multi-digit intermediate sum and multi-digit carry-over.

Rule 5. The multi-digit intermediate sum is reduced to the MINIMAL FORM and then is summarized with the multi-digit carry-over.

Rule 6. The summation process continues in accordance with the rules 4, 5 until obtaining the multi-digit carry-over equal to 0. The last intermediate sum, reduced to the MINIMAL FORM is the result of Fibonacci summation.

For the above Fibonacci summation we need to add the following additional rule.

Rule 7. Consider the case when we have two binary 1's in the k^{th} digits of the summand Fibonacci representations. It follows from the property of the MINIMAL FORM that the bits of the (k+1)-th and (k-1)-th digits of the both summand Fibonacci representations are always equal to 0. It is clear that for this case the intermediate sums, arising at the addition of the (k+1)-th and (k-1)-th digits of the both summand Fibonacci representations, are always equal to 0. This means that we can place one of the carry-over's, arising at the summation of the k-th significant digits (1+1), at once to the (k-1)-th digit of the intermediate sum (for the (a)-method of the Fibonacci summation) or to the (k+1)-th digit of the intermediate sum (for the (b)-method of the Fibonacci summation).

We can demonstrate the above summation rules by the following example.

Example 3. Fibonacci summation of multi-digit Fibonacci representations

Sum the Fibonacci representations 31=10011011 and 22=01011010, represented in the Fibonacci code (9):

1. Reduction of the summand Fibonacci representations to the MINIMAL FORM:

$$31 = 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0$$
$$22 = 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0.$$

2. Formation of the multi-digit intermediate sum S_1 and multi-digit carry-over C_1 in accordance with the method (a) of Table 7:

$$\begin{array}{l} 31 = 10100100 \\ \underline{22 = 10000010} \\ S_1 = 11100110 \\ C_1 = 00100000 \end{array}$$

3. Reduction of the intermediate sum S_1 to the MINIMAL FORM:

$$S_1 = 100101000$$

4. Summation of S_1 and C_1:

$$\begin{array}{l} S_1 = 100101000 \\ \underline{C_1 = 000100000} \\ S_2 = 100111000 \\ C_2 = 000001000 \end{array}$$

5. Reduction of the intermediate sum S_2 to the MINIMAL FORM:

$$S_2 = 101001000$$

6. Summation of S_2 and C_2:

$$\begin{array}{l} S_2 = 101001000 \\ \underline{C_2 = 000001000} \\ S_3 = 101001100 \\ C_3 = 000000010 \end{array}$$

7. Reduction of the intermediate sum S_3 to the MINIMAL FORM:

$$S_3 = 101010000$$

8. Summation of S_3 and C_3:

$$\begin{array}{l} S_3 = 101010000 \\ \underline{C_3 = 000000010} \\ S_4 = 101010010 \\ C_4 = 000000000 \end{array}$$

The summation is over because carry-over C_4 =0.

As it is well-known, the method of the «direct» number subtraction in the classical binary arithmetic (p = 0) is based on the following property of the binary numbers:

$$2^{n+k} - 2^n = 2^{n+k-1} + 2^{n+k-2} + ... + 2^n. \quad (39)$$

Write now the similar identity for the Fibonacci (p=1)-numbers:

$$F_{n+k} - F_n = F_{n+k-2} + F_{n+k-3} + ... + F_{n-1}. \quad (40)$$

Using the identity (40) and Fibonacci recurrence relation (10), we can construct the following Fibonacci subtraction table:

Table 8. Fibonacci subtraction table for the case *p*=1

0	-	0	=	0
1	-	1	=	0
1	-	0	=	0 1 1
1 0	-	1	=	0 1
1 0 0	-	1	=	1 1
1 0 0 0	-	1	=	1 1 1

The direct Fibonacci subtraction of multi-digit numbers uses the following rules:

Rule 8. Before subtraction the subtracted Fibonacci representations are reduced to the MINIMAL FORM.

Rule 9. The subtracted MINIMAL FORMS are compared by their value according to the rules of Fibonacci representations comparison and then in accordance with Table 8, the lesser Fibonacci representation is subtracted from the bigger Fibonacci representation.

Note that these arithmetic algorithms are new technical solutions. They were developed by the author in the early 70s of the 20th century and these algorithms have been used as a basis of comparison devices, addition and subtraction devices, as well as summing and subtracting Fibonacci counters, patented abroad [16-28].

It should not be assumed that the above rules of arithmetical operations in the Fibonacci code (9) is the only possible rules. In the book [31], the rules of Fibonacci subtraction based on the use of the concepts of additional and inverse Fibonacci codes are developed. In the same book [31] the original methods of the Fibonacci summation and subtraction, based on the use of the so-called "basic micro-operations," are suggested.

4.4 Fibonacci multiplication and division

4.4.1 Fibonacci multiplication

The analysis of the *Ancient Egyptian multiplication* [46] led us to the following method of the *p-Fibonacci multiplication.*

Consider now the product $P = A\times B$, where the numbers A and B are represented in the Fibonacci p-code (6). Using the representation of the multiplier B in the Fibonacci p-code (6), we can write the product $P = A\times B$ in the following form:

$$P = A\times b_n F_p(n) + A\times b_{n-1} F_p(n\text{-}1) + \ldots + A\times b_i F_p(i) + \ldots + A\times b_1 F_p(1), \tag{41}$$

where $F_p(i)$ is the Fibonacci p-number.

The following algorithm of the *p-Fibonacci multiplication* follows from the expression (41). The multiplication is reduced to the summation of the partial products of the kind $A\times b_i F_p(i)$. They are formed from the multiplier A according to the special procedure that reminds the *Ancient Egyptian multiplication* [46]. Demonstrate now the "Fibonacci multiplication" for the case of the simplest Fibonacci p-code (p=1).

Example 4. Example of Fibonacci multiplication: 41×305.

1. Construct the table consisting of the three columns: *F, G* and *P* (see Table 9).
2. Insert the Fibonacci 1-sequence (the classical Fibonacci numbers) 1, 1, 2, 3, 5, 8, 13, 21, 34 to the *F*-column of Table 9.
3. Insert the generalized Fibonacci 1-sequence: 305, 305, 610, 915, 1525, 2440, 3965, 6505, 10370, which is formed in the *G*-column from the first multiplier 305 according to the "Fibonacci recurrence relation."
4. Mark by the inclined line (/) and fat all the *F*-numbers that give the second multiplier in the sum (41 = 34 + 5 + 2).
5. Mark by fat all the *G*-numbers corresponding to the marked *F*-numbers and rewrite them to the *P*-column.
6. Summarizing all the *P*-numbers, we obtain the product: 41×305 = 12 505.

Table 9. Fibonacci multiplication for the case *p*=1

F	*G*	*P*
1	305	
1	305	
/2	**610**	→ **610**
3	915	
/5	**1 525**	→ **1 525**
8	2 440	
13	3 965	
21	6 505	
/34	**10 370**	→ **10 370**
41=34+5+2	**41×305**	= **12 505**

The above Fibonacci multiplication algorithm is easily generalized for the case of the Fibonacci p-codes (6).

<u>**4.4.2 Fibonacci division**</u>

Consider the **Example 5** of the Fibonacci division for the case p=1.

Example 5. Divide the number 481 (the dividend) by the number 13 (the divisor) in the Fibonacci code (9) (*p*=1).

1. Construct the table consisting of three columns: *F, G* and *D* (see Table 10).
2. Insert the Fibonacci 1-sequence (the classical Fibonacci numbers) 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 to the *F*-column of Table 10.

3. Insert the generalized Fibonacci 1-sequence: 13, 13, 26, 39, 65, 104, 169, 273, 442, 615, formed from the divisor 13 according to the "Fibonacci recurrence relation," to the *G*-column.
4. Compare sequentially every *G*-number with the dividend 481, inscribed into the *D*-column, and fix the result of comparison (≤ or >) until when we obtain the first comparison result of the kind (>): 615>481.
5. Mark by the incline line (/) and fat the *F*-number **34**, corresponding to the preceding *G*-number 442, and mark the letter by fat.
6. Calculate the difference: R_1 = 481 – 442 = 39.

Table 10. The first stage of the Fibonacci division

F	*G*	*D*
1	13	≤481
1	13	≤481
2	26	≤481
3	39	≤481
5	65	≤481
8	104	≤481
13	169	≤481
21	273	≤481
/34	**442**	≤481
55	615	>481
$\boldsymbol{R_1}$ **= 481 – 442 = 39**		

The second stage of the Fibonacci 1-division is a repetition of the first stage but we use instead the dividend 481 the difference R_1 = 39 (see Table 11).

Table 11. The second stage of the Fibonacci division

F	*G*	*D*
1	13	≤39
1	13	≤39
2	26	≤39
/3	**39**	≤39
5	65	>39
$\boldsymbol{R_2}$ **= 39 – 39 = 0**		

Because the second difference R_2 = 39 – 39 = 0, this means the 1-Fibonacci division is over. The result of the division is equal to the sum of all the fatted *F*-numbers obtained on all stages (see Tables 10, 11), that is:

481 : 13 = 34 + 3 = 37

5 Original Fibonacci Devices

5.1 The device for reduction of Fibonacci representations to MINIMAL FORM

The "convolution" and "devolution" devices play an important role in the technical realization of arithmetical operations over the Fibonacci representations. They can be designed on the base of the binary register having special logical circuits to perform "convolutions" and "devolutions". Each digit of the register contains binary flip-flop" (trigger) and logical elements. The operations of "convolution" (011 = 100) and "devolution" (100 = 011) can be performed by means of the inversion of the "flip-flops" (triggers).

One of the possible variants of the "convolution" register or "the device for reduction of the Fibonacci code to the minimal form" is shown in Fig. 2.

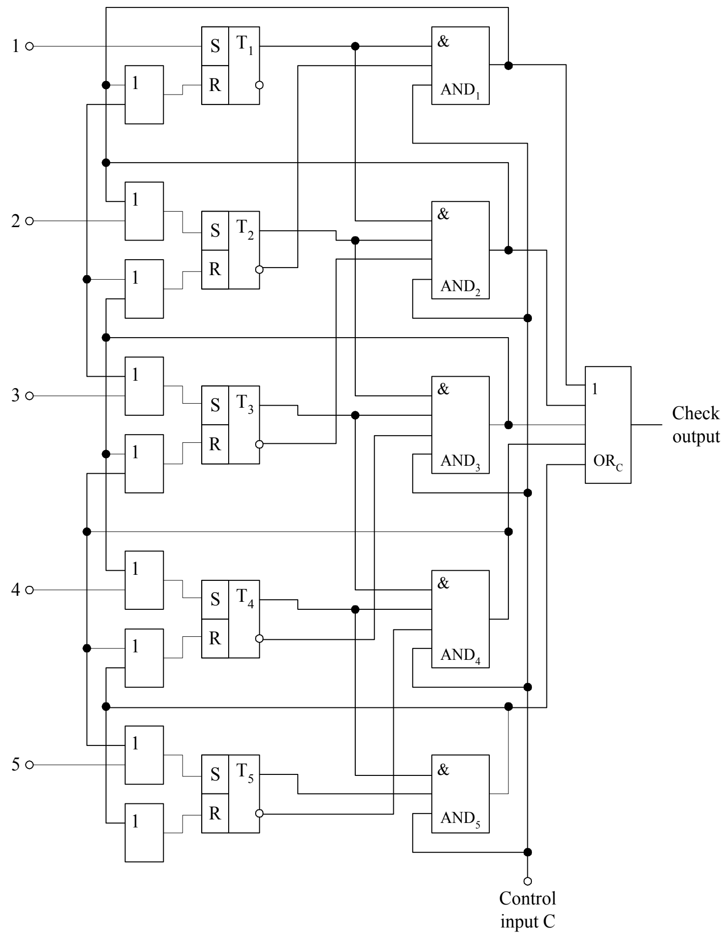


Fig. 2. The device for reduction of the Fibonacci code to the MINIMAL FORM

The device in Fig. 1 consists of the five *R-S*-triggers and the logical elements *AND*, *OR,* which are used to perform the "convolutions". The "convolution" is performed by using the logical gates AND_1 - AND_5 and corresponding logical gate *OR,* standing before the *R*- and *S*-inputs of the triggers. The logical gate AND_1 performs the "convolution" of the 1-st digit to the 2-d digit. Its two inputs are connected with the direct output of the trigger T_1 and the inverse output of the trigger T_2. The 3-d input is connected with the synchronization input *C*. The logical gate AND_1 analyzes the states Q_1 and Q_2 of the triggers T_1 and T_2. If $Q_1 = 1$ and $Q_2 = 0$, this means that the convolution condition is satisfied for the 1-st and 2-d digits. The synchronization signal $C = 1$ causes the appearance of the logical 1 at the output of the gate AND_1 . The latter causes switching the triggers T_1 and T_2. This results to the "convolution" (01 = 10).

The logical gate AND_k of the *k*-th digit (*k*=2, 3, 4, 5) performs the "convolution" of the (*k*-1)-th and *k*-th digits to the (*k*+1)-th digit. Its three inputs are connected with the direct outputs of the triggers T_{k-1} and T_k and the inverse output of the trigger T_{k+1}. The 4-th input is connected with the synchronization input *C*. The logical gate AND_k analyzes the states Q_{k-1}, Q_k, and Q_{k+1} of the triggers T_{k-1}, T_k, and T_{k+1}. If $Q_{k-1} = 1$, $Q_k = 1$, and $Q_{k+1} = 0$, this means that the "convolution" condition is satisfied. The synchronization signal $C = 1$

results switching triggers T_{k-1}, T_k, and T_{k+1}. The "convolution" of the corresponding digits (011 = 100) is over.

Notice that all the gates AND_1 - AND_5 are connected through the common logical gate OR_c with the check output of the "convolution" register.

The device for bringing of the Fibonacci code to the MINIMAL FORM in Fig. 1 operates in the following manner. The input code information is sent to the information inputs 1 - 5 of the "convolution" register and enters the *S*-inputs of the triggers through the corresponding logical gates *OR*. Let the initial slate of the convolution register be in the following state:

5 4 3 2 1
0 1 0 1 1

It is clear that the "convolution" condition is satisfied only for the 1-st, 2-d and 3-d digits. The first synchronization signal $C = 1$ results in the passage of the "convolution" register to the following state:

5 4 3 2 1
0 1 1 0 0.

Here the "convolution" condition is satisfied for the 3-d, 4-s and 5-th digits. The next synchronization signal $C = 1$ results in the passage of the "convolution" register to the following state:

5 4 3 2 1
1 0 0 0 0.

The "convolution" is over.

5.2 The "convolution" register as self-checking device

The outputs of the logical gates AND_1 - AND_5 of the "convolution" registers in Fig. 1 are connected with the register check output through the common gate *OR*. This output plays an important role as the check output of the "convolution" register.

It follows from the functioning principle of the "convolution" register that the logical 1 appears on the check output only for two situations:

(1) The binary code word, written into the "convolution" register, is not MINIMAL FORM. This means that the "convolution" condition is satisfied at least for one triple of the adjacent triggers of the "convolution" register. This causes the appearance of the logical 1 at the output of the corresponding gate *AND*. Hence in this case the appearance of the logical 1 at the check output of the "convolution" register indicates the fact that the "convolution" process is not over. This means that we have a possibility to indicate the termination of the "convolution" process by means of observing the check output of the "convolution" register.
(2) The appearance of the constant logical 1 at the check output is an indication of the fault in the "convolution" register. Hence the "**convolution" register is a natural self-checking device** what is important for the improvement of informational reliability of Fibonacci computer.

There are other variants of implementation of such device, described in the book [31].

5.3 Device for checking MINIMAL FORM

Fig. 3 shows a device for checking MINIMAL FORM. The device consists of the *n* logical gates *AND*. Their outputs are connected with the inputs of the common logic gate *OR*. If the initial Fibonacci representation has a violation of the MINIMAL FORM, that is, has two adjacent bits of 1 or the bit 1 in the lower digit, there appears the logical 1 at least at one input of the logical gate *AND*. It results in the appearance of the logical 1 at the output of the common gate *OR* and this logical 1 is the error indication.

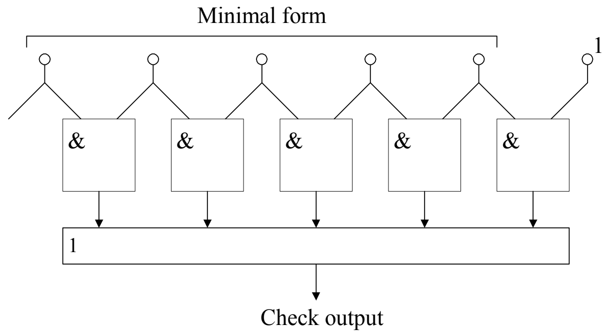


Fig. 3. The logical circuit for checking the minimal form

6 Conception of Fibonacci Computer and Its Advantages

6.1 The main principles of Fibonacci computer

We can formulate the following main principles of the Fibonacci computer:

1. We use in Fibonacci computer the redundant Fibonacci *p*-codes. However, **Fibonacci (*p*=1) –code**, having the least code redundancy, is the **most suitable** in terms of hardware cost.
2. We use the **MINIMAL FORM** of the Fibonacci (*p*=1)–code for error detection at all stages of data transformation, including arithmetic operations, data transfer and storage.

Fibonacci computer has a number of important advantages in comparison with classic binary computers. The most important of these is the high ability to detect errors in functional units of Fibonacci computer.

6.2 Error detection

6.2.1 "Soft" and "hard" errors

As it is well-known, all errors, arising in functional devices of computer, can be divided into two groups: 1) the *"soft errors"* that result from random effects on electronic elements and the *"hard errors"* that result from constant failures of electronic elements. Both types of errors are dangerous and may lead to "false" data on the computer.

As for the *"hard errors"*, they can be detected by the register for reduction of Fibonacci code to the MINIMAL FORM. This register is an important device of all arithmetic units and thanks to this device all Fibonacci arithmetic devices become self-checking devices. This is the first important advantage of the Fibonacci computer.

6.2.2 Formula for error-detecting ability for "soft errors"

Let us consider now the error detection in such an important computer unit as an electronic memory. Error-detecting ability of the Fibonacci *p*-codes is determined by the relationship between the *allowed* and *forbidden* code combinations. Note that the set of the *allowed* combinations of the *n*-bit Fibonacci *p*-code coincides with the set of the *n*-bit MINIMAL FORMS. This means that the number of the *n*-bit *allowed* combinations is equal to $N_1 = F_p(n+1)$. Then, the number of the *forbidden* code combinations is equal to $N_2 = 2^n - F_p(n+1)$, where 2^n is the number of the possible n-bit binary combinations. Then the **ERROR-DETECTING ABILITY** S_d of the Fibonacci p-code (for the case of the MINIMAL FORM), is determined as follows:

$$S_d(p) = \frac{N_2}{2^n} = 1 - \frac{F_p(n+1)}{2^n}. \tag{42}$$

For example, for the case of the 24-bit Fibonacci p-codes (p=1 and p=2) we have the following numerical values for the range of representations, respectively:

$$\boldsymbol{p=1}: F_1(25) = 62215,\ 2^{24} = 16777216 \tag{43}$$

$$\boldsymbol{p=2}: F_1(25) = 6450,\ 2^{24} = 16777216. \tag{44}$$

By using (42) - (44), we can calculate the ERROR-DETECTING ABILITY of the 24-bit Fibonacci 1- and 2-codes, respectively:

$$\begin{aligned} S_d(p=1) &= 0.9963(99.63\%) \\ S_d(p=2) &= 0.9996(99.96\%) \end{aligned} \tag{45}$$

<u>6.2.3 Fibonacci parity code</u>

As seen from the example (45), the potential error detecting ability of the Fibonacci p-codes is enough high. In order to improve the potential error-detection ability of the Fibonacci code, we can use the so-called Fibonacci Parity Code (FPC) by adding a PARITY BIT a_{par} to the MINIMAL FORM (MF) of the Fibonacci code:

$$\underbrace{a_n a_{n-1} \ldots a_i \ldots a_2 a_1}_{MF} \underbrace{a_{par}}_{PB} \tag{46}$$

The FIBONACCI PARITY CODE (46) significantly improves the error-detecting ability of the Fibonacci p-code. In this case, the main feature of the FPC is ensuring the 100% detection of all odd-bit errors, in particular, the single-bit errors. It is easy to prove [31] that the POTENTIAL ERROR-DETECTING ABILITY of the FPC (46) is calculated by the formula:

$$S_d(FPC) = 1 - \frac{F_p(n+1)}{2^{n+1}}. \tag{47}$$

For example, for the case of the 24-bit Fibonacci p-codes (p=1 and p=2) we have the following data ranges for the FIBONACCI-PARITY CODES (p=1 and p=2), respectively:

$$F_1(25) = 62215,\ 2^{25} = 33554432 \tag{48}$$

$$F_2(25) = 6450,\ \ 2^{25} = 33554432. \tag{49}$$

By using (47) for the cases (48), (49), we can calculate the ERROR-DETECTING ABILITY of the FIBONACCI-PARITY CODES (p=1 and p=2), respectively, as follows:

$$S_d\left(FPC, p=1\right)=0.9981459\left(99.8\%\right) \tag{50}$$

$$S_d\left(FPC, p=2\right)=0.9998078\left(99.98\%\right). \tag{51}$$

This means that the FIBONACCI-PARITY CODE can provide the continuous detection of errors in Fibonacci microcontroller or microprocessor at various stages of storage, transmission and processing of data with the error detection coefficient equal to 99.8-99.98%.

<u>**6.2.4 Energy consumption and power dissipation in ROM**</u>

It is known that for certain types of electronic memory (EM) there is some asymmetry between bits 1 and 0 at their storage in different kinds of electronic memory, for example, ROM. In particular, the recording of the bit 1 and its reading from the ROM requires more energy consumption, than for the bit 0. From this point of view, the MINIMAL FORM of the Fibonacci representations is an optimal binary representations from the point of view of **energy consumption and power dissipation**, because the bits 1 are separated always with bits 0 (in general, in the MINIMAL FORM, two bits 1 are separated by no less then p bits 0).

It is clear, in the array of the MINIMAL FORMS, the bits 1 and 0 are distributed non-uniformly; in this case, always the number of the bits 0 exceeds the number of the bits 1. This creates "comfortable" conditions for electronic memory, in particular, ROM, from the point of view of the **energy consumption and power dissipation**. The ROM's with fusible links or electrically programmable ROM's are examples of such kind of electronic memory. For such ROM's only the bits 1 determine energy consumption and power dissipation.

In order to estimate the decrease of energy consumption in ROM, when the data is stored in the MINIMAL FORM of the Fibonacci p-codes (6), we consider the following reasoning's. When we store in ROM the numbers, represented with the m-bit binary code, the maximal energy consumption appears at recording and reading of the following binary code combination:

$$N_{\max}\left(m\right)=\underbrace{11...1}_{m}. \tag{52}$$

If the storage of one bit of 1 demands the energy consumption P_1, then we can express the maximal energy consumption $P_{\max}\left(m\right)$ for the storage of the code combination (65) as follows:

$$P_{\max}\left(m\right)=mP_1 . \tag{53}$$

With the m-bit binary code we can represent 2^m positive integers, that is, the range of the number representation is equal:

$$D=2^m. \tag{54}$$

If we will represent the positive integers in the MINIMAL FORM of the Fibonacci p-code (6), then, due its code redundancy, for the storage of the same range of positive integers, given by (54), we need to increase the number of digits of the Fibonacci p-code (6) proportionally to its relative code redundancy r. In this case, the number of the Fibonacci's bits n for the storage of the number range (54) is equal approximately to:

$$n=m\left(1+r\right). \tag{55}$$

The formula (5568) shows that the number of the bits n in the Fibonacci p-code (6) increases by $(1+r)$ times in comparison with the number of the bits m for the binary system, necessary for the representation of the same number range (54). In this case, the n-bit Fibonacci representation of the maximal number $N_{\max}$ in the MINIMAL FORM of the Fibonacci p-code (6) looks as follows:

$$N_{\max} = 1\underbrace{00...0}_{p}1\underbrace{00...0}_{p}...1\underbrace{00...0}_{p}...1\underbrace{00...0}_{p}. \quad (56)$$

In particular, for the case p=1 we have:

$$N_{\max} = 1010...10...10. \quad (57)$$

We can see that the Fibonacci representation (56) consists of the k groups of the kind $1\underbrace{00..0}_{p}$, which contains one bit of 1 and p bits of 0, where k is the number of the bits of 1 in the representation (56). By using (55) and (56), we can express the number of k as follows:

$$k = \frac{n}{p+1} = \frac{m(1+R)}{p+1}. \quad (58)$$

It follows from (58) that the maximal energy consumption for the storage of the n-bit Fibonacci representation is determined by the expression:

$$P_{\max}^{f} = kP_1 = \frac{m(1+r)}{p+1}P_1. \quad (59)$$

Consider now the ratio:

$$\beta = \frac{P_{\max}}{P_{\max}^{f}} = \frac{p+1}{1+R}. \quad (60)$$

Because R<1 and $p \geq 1$, then the coefficient β describes the decrease of the energy consumption in ROM, if we use the Fibonacci p-codes (6).

Table 12 sets forth the values of β for the cases p=1, 2.

Table 12. The values of β

p	1	2
R	$\approx 0.33(33\%)$	$\approx 0.5(50\%)$
β	1.5	2

Thus, for the case p=2 the improvement in energy consumption is about 2 times, despite the fact that the number of digits in Fibonacci 2-code increases by about 1.5 times compared with the classical binary system. It follows from Table 12 that the gain in the energy consumption in the ROM increases with increasing p. This result could have great significance for the future of nano-computers and microelectronics, where decreasing energy consumption and optimizing power dissipation is becoming one of the central problems.

6.2.5 Fibonacci *p*-codes as self-synchronization codes

A distinction of the bits 1 and 0 in sequentially transmitted binary data relates to a serious technical problem, called *problem of synchronization*. To solve this problem, special synchronization signals, *clock-signals*, need to be used. One of the effective ways for improving of the synchronization of transmitted data (without special clock-signals of synchronization) is to use self-synchronisation codes (see diagram B in Fig. 4).

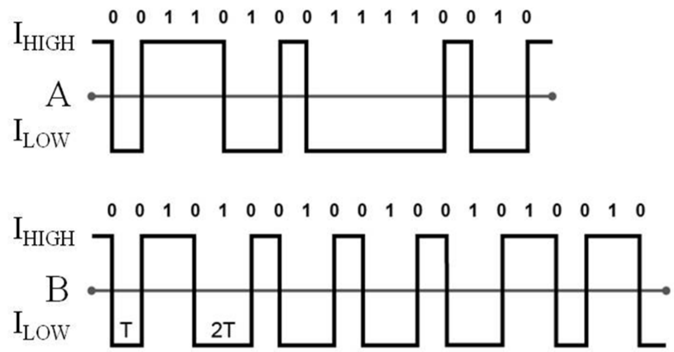


Fig. 4. Fibonacci code as self-synchronization code

The diagrams A, B represent a widespread method transmitting binary data. Switching two electrical currents performs the transmission of bits: the high level current I_{high} and the low level current I_{low}. The switching from one level to another is performed only for the bit 0; the switching is not performed for the bit of 1. The diagram A shows the electrical signal, which is generated during the transmission of the classic binary code combination without self-synchronization mechanism. The greatest difficulty in the transmission of the signal arises when the transmitted binary sequence contains a long "packet," consisting of bits 1.

The problem of distinction of bits 1 and 0 in this method is greatly simplified if we impose certain restrictions on the length of the "packets," consisting of the consecutive bits of 1. The simplest solution is to use the codes, in which two bits of 1 do not appear together. This restriction is a basic property for MINIMAL FORMS of all the Fibonacci *p*-codes (20). For the first time, American engineer **W.H. Kautz** paid attention to such use of the Fibonacci code for synchronization control [47].

The diagram B shows the example of the formation of the digital signal by using simplest Fibonacci code (p=1). Denote by T the period of the original code sequence. Analysis of the digital signal, generated from the MINIMAL FORM of the Fibonacci code (p=1), shows that this digital signal contains only two pulse durations T and $2T$. This digital signal is called *dual-frequency signal*. It is easy to form clock-signals from the dual-frequency signal. This fact is a confirmation of the self-synchronization property of the Fibonacci *p*-codes (p=1,2,3,…).

7 Codes of the Golden *p*-proportions, Bergman's System and Their Applications

7.1 Definition of the codes of the golden *p*-proportions

The Fibonacci *p*-codes (6) are intended for representation of natural numbers. In author's publications [6-8,11,15], the so-called *codes of the golden p-proportions*, intended for positional representation of real numbers, have been introduced.

Let us consider the binary code for real numbers:

$$A = \sum_i a_i 2^i \left(i = 0, \pm 1, \pm 2, \pm 3, \ldots\right), \tag{61}$$

where the digit weights $2^i \left(i = 0, \pm 1, \pm 2, \pm 3, \ldots\right)$ are connected with the following well-known identities:

$$2^i = 2^{i-1} + 2^{i-1} \left(summing\ identity\right) \tag{62}$$

$$2^i = 2 \times 2^{i-1} \left(multiplicative\ identity\right) \tag{63}$$

The binary code (61) allows the following generalization. Consider the set of the following standard line segments:

$$\{\ldots, \Phi_p^n, \Phi_p^{n-1}, \ldots, \Phi_p^{n-p-1}, \ldots, \Phi_p^0 = 1, \Phi_p^{-1}, \ldots, \Phi_p^{-k}, \ldots\} \tag{64}$$

where $\Phi_p \left(p = 0, 1, 2, 3, \ldots\right)$ are the golden *p*-proportions, which are real roots of the golden *p*-ratio equation (5). The powers of the golden *p*-proportions $\Phi_p^n \left(p = 0, 1, 2, 3, \ldots; n = 0, \pm 1, \pm 2, \pm 3, \ldots\right)$ are connected by the remarkable identities:

$$\Phi_p^n = \Phi_p^{n-1} + \Phi_p^{n-p-1} \left(summing\ identity\right) \tag{65}$$

$$\Phi_p^n = \Phi_p \times \Phi_p^{n-1} \left(multiplicative\ identity\right) \tag{66}$$

By using (64), we can obtain the following positional binary method of real numbers representation called the *code of the golden p-proportions* [6-8,11,15,31]:

$$A = \sum_i a_i \Phi_p^i;\ p = 0, 1, 2, 3, \ldots;\ i = 0, \pm 1, \pm 2, \pm 3, \ldots, \tag{67}$$

where $a_i \in \{0, 1\}$ is the bit of the *i*-th digit; Φ_p^i is the weight of the *i*-th digit; Φ_p is the base of the numeral system (67).

The abridged notation of any real number *A* for the code of the golden *p*-proportion (67) is called the *"golden" representation* and has the following form:

$$A = a_m a_{m-1} \ldots a_1 a_0, a_{-1} a_{-2} \ldots a_{-(k-1)} a_{-k} \ldots \tag{68}$$

Note that the formula (67) gives an infinite number of new positional numeral systems with *irrational bases*, since all their bases $\Phi_p \left(p = 1, 2, 3, \ldots\right)$ are irrational numbers (excepting for the base $\Phi_{p=0} = 2$, corresponding to the case *p*=0).

Note that the *codes of the golden p-proportions* (67) for the first time were introduced by the author in the 1978- and 1980-articles [6-8]. Theory of the codes of the golden *p*-proportions and their applications at *digital metrology* and *computer science* is described in author's 1978-articles [6,7] and 1984- and 2009-books [15,31].

7.2 Bergman's system

Let us consider the partial cases of the sum (67). It is clear that for the case p=0 $\Phi_{p=0}=2$ and we get the classical binary representation of real numbers, given with (61).

For the case p=1, we get very unexpected result. Really, for this case we have:

$$\Phi_{p=1}=\Phi=\frac{1+\sqrt{5}}{2}\left(the\ golden\ ratio\right) \tag{69}$$

and we get from (67) and (69) the following remarkable expression:

$$A=\sum_i a_i\Phi^i=\sum_i a_i\frac{1+\sqrt{5}}{2}\left(i=0,\pm1,\pm2,\pm3,...\right)\ , \tag{70}$$

where the powers of the golden ratio $\Phi^n\left(n=0,\pm1,\pm2,\pm3,...\right)$ are connected with the following identities:

$$\Phi^n=\Phi^{n-1}+\Phi^{n-2}\left(summing\ identity\right) \tag{71}$$

$$\Phi^n=\Phi\times\Phi^{n-1}\left(multiplicative\ identity\right). \tag{72}$$

The unexpectedness of the result (70) consists of the following. The formula (70) was obtained for the first time in 1957 by the American mathematician **George Bergman,** who published this result in the article *A number system with an irrational base* [48].

On the first view, there aren't particular distinctions between the formula (70) for Bergman's system and the formula (61) for the binary system. However, it is only on the first view.

The principal distinction of the numeral system (70) from the binary system (61) consists of the fact that the irrational number $\Phi=\left(1+\sqrt{5}\right)/2$ (the golden ratio) is used as the base of the numeral system (70). Although Bergman's article [48] presents a result of fundamental importance for number theory and computer science, the mathematicians and experts in computer science did not take notice to Bergman's system in that period. They simply ignored this mathematical discovery, which predicted a new direction in the development of computer science called the *"Golden" Computers.* In the conclusion of his paper [48] **George Bergman** wrote the following pessimistic conclusion: *"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."*

Analyzing Bergman's system (70) and comparing it to the codes of the golden p-proportions (67) by using historical-mathematical approach, we come to the following conclusions:

1. Bergman's system (70) is the **first in the mathematical literature numeral system with an irrational base** (the golden ratio).
2. Bergman's system is possibly the **most important mathematical discovery in the field of numeral systems after the discoveries of the positional principle of number representation (Babylon, c. 2000 B.C.E.) and the decimal system (India, 5th century AD).** The importance of Bergman's system for the development of numeral systems can be compared with the introduction of irrational numbers by Pythagoreans in Ancient Greece.

3. Bergman's system (79) and its generalization, codes of the golden p-proportions [6, 8], return mathematics to its historical sources (Babylon, Ancient Egypt), when numeral systems were in the center of mathematics and determined its content [46].
4. The most surprising conclusion consists of the fact that **George Bergman** made his mathematical discovery at the age of 12! **This case is unprecedented in the history of mathematics!**
5. The *codes of the golden p-proportions* (67), introduced and studied by the author in the works [6-8,11,15,31], are generalization of the two great mathematical achievements:

   1) On the one hand, the *codes of the golden p-proportions* are generalization of the *binary system* (61), which is the basis of modern computer science.
   2) On the other hand, they are generalization of the Bergman's system (70) as new definition of *real numbers* what has fundamental importance for the development of number theory and mathematics.

6. Notice that a number of the *codes of the golden p-proportions* is theoretically infinite because every $p\left(p=0,1,2,3,...\right)$ "generates" its own positional binary (0,1) representation of the kind (67). Excepting of the case p=0, all other bases of numeral systems (67) are irrational. This means that the expression (67) defines an infinite number of numeral systems with irrational bases. The traditional binary system (61) (p=0) with the base of 2 is the only exception from the new class of numeral systems with irrational bases, given with (67).

The following directions of applications of Bergman's system and codes of the golden p-proportions follow from the above conclusions:

1. "Golden" number theory [34].
2. "Golden" arithmetic for computer science [15,31].
3. "Golden" ternary mirror-symmetrical arithmetic [29].
4. "Golden" self-correcting digital metrology [6].

Let us consider the above "golden" applications more in detailed.

7.3 A conception of the "golden" number theory

The detailed description of this concept is given in 2015 article [34] and the book [31]. As it is well-known, the *elementary number theory,* described in Euclid's *Elements,* begins from Euclidean definition of *natural numbers*, given with (11).

The "golden" number theory [31,34] treats Bergman's system (70) and codes of the golden p-proportions (67) as a new definition of real numbers. Such approach leads us to new unexpected properties of natural numbers, proved in [31,34]:

1. The "golden" representation (67) for any natural number N in Bergman's system (70) and codes of the golden p-proportions (67) has always finite number of digits that is far not trivial.
2. Z-property [31,34]:

$$\begin{aligned}&For\ any\ N=\sum_i a_i\Phi^i\ after\ substitution\ F_i\to\Phi^i\\&we\ get:\sum_i a_iF_i\equiv 0\left(i=0,\pm1,\pm2,\pm3,...\right)\end{aligned} \tag{73}$$

Therefore, the above properties 1 and 2 and other properties (D-property, F-code and L-code), described in [31,34], are true only for natural numbers and can be considered as **new properties of natural numbers**. This means that the conception of the **"golden" number theory** [31,34] led us to previously unknown properties of natural numbers, the theoretical study of which began 2.5 millennia ago, at least starting from

Euclid's *Elements*. These properties are of great interest for number theory and can be used in computer science.

7.4 "Golden" arithmetic for computer science

Comparing the properties of the codes of the golden *p*-proportions (67), given by (65) and (66), and Bergman's system (70), given by (71), (72), to the properties of the Fibonacci *p*-codes, given by (1), (2) and (10), which determine arithmetical properties of the Fibonacci *p*-codes (6), we conclude that Bergman's code (70) and the codes of the golden *p*-proportions (67) are similar to Fibonacci *p*-codes (6). This leads s to the conclusion that the "golden" arithmetic is similar to the Fibonacci arithmetic and can use the same technical solutions, in particular, the same device for the reduction of the Fibonacci and "golden" representations to the MINIMAL FORM (Fig. 1) and the same device for checking MINIMAL FORM (Fig. 2).

On the other hand, comparing the properties (71), (72) of Bergman's code (70) and the properties (65), (66) of the codes of the golden *p*-proportions (67) to the properties (62), (63) of the classic binary code (61), we conclude that the "golden" arithmetic is similar to the classical binary arithmetic when performing certain arithmetic operations, such as multiplication and division.

A detailed description of the "golden" arithmetic is given in the book [31].

8 "Golden" Ternary Mirror-symmetrical Representation and Arithmetic

8.1 Brief introduction

In 2002 "The Computer Journal" (British Computer Society) has published author's article *Brousentsov's ternary principle, Bergman's number system and ternary mirror-symmetrical arithmetic* [29]. As follows from the title of the article [29], the main purpose of the article [29] was to develop *ternary mirror-symmetrical arithmetic* based on *Bergman' system* [45] and *ternary principle,* used by Russian engineer **Nikolay Brousentsov** (1925 —2014) for designing the first in computer history ternary computer "Setun" (Moscow University). The article [29] caused a positive reaction from the Western computer community. The prominent American mathematician and a world-renowned expert in computer science **Donald Ervin Knuth** (see Fig. 5) was the first outstanding scientist who congratulated the author with the publication of the article [29]. In his letter, he informed the author about his intention to include a description of the *"golden" mirror-symmetrical arithmetic* into the new edition of the book *"Art of Computer Programming."*

Fig. 5. Donald Ervin Knuth

American mathematician and expert in computer science, author of the world-known bestseller "Art of Computer Programming" (1968, 1969, and 1973).

Why Prof. Donald Knuth so interested in the article [29]? A detailed answer to this question is given in the article [29] and the author's book [31]. Below we present the most common ideas of new computer arithmetic.

8.2 "Golden" ternary mirror-symmetrical representation

8.2.1 Conversion of Bergman's code to the ternary "golden" mirror-symmetrical representation

As it is mentioned above, any positive integer *N* has a unique "golden" representation in the MINIMAL FORM. This means that each bit a_k=1 in the binary "golden" representation of *N* would be "enclosed" by the two next bits $a_{k-1} = a_{k-2} = 0$.

Consider now the following identity for the powers of the golden ratio Φ :

$$\Phi^k = \Phi^{k+1} - \Phi^{k-1} \tag{74}$$

The identity (74) has the following code interpretation:

$k+1$	k	$k-1$		$k+1$	k	$k-1$
0	1	0	=	1	0	$\bar{1}$

(75)

where $\bar{1}$ is the negative unit, that is, $\bar{1}$ = - 1. It follows from (75) that the bit 1 of the k^{th} digit is transformed into two 1's, the positive unit 1 of the $(k+1)^{th}$ digit and the negative unit $\bar{1}$ of the $(k-1)^{th}$ digit.

It is proved in [29] that any integer *N* (positive or negative) may be represented as follows:

$$N = \sum_i b_i \Phi^{2i} \ , \tag{76}$$

where $b_i \in \{\bar{1}, 0, 1\}$ is a ternary numeral (treat), Φ^{2i} is the weight of the i th digit, $\Phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio.

8.2.2 Ternary "golden" representations

The abridged notation of the sum (76):

$$N = b_k b_{k-1} \ldots b_2 b_1 b_0 , b_{-1} b_{-2} \ldots b_{-(k-1)} b_{-k} \tag{77}$$

is called *ternary "golden" representation* of integer *N.* Table 13 shows the *ternary "golden" representations* of the integers *N* in the range {0,1,2,…,10}. We use in the *ternary "golden" representations* of the Table 13 the 7 ternary digits $i \in \{3, 2, 1, 0, -1, -2, -3\}$ of the kind $\{\bar{1}, 0, 1\}$.

Table 13. Ternary "golden" representations

i	3	2	1	0	-1	-2	-3
Φ^{2i}	Φ^6	Φ^4	Φ^2	Φ^0	Φ^{-2}	Φ^{-4}	Φ^{-6}
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

8.2.3 Property of "mirror symmetry"

Studying the ternary "golden" representations of Table 13, we find important regularity for all ternary "golden" representations (77). If we compare the left $\{b_k b_{k-1} \ldots b_2 b_1\}$ and right $\{b_{-1} b_{-2} \ldots b_{-(k-1)} b_{-k}\}$ parts of any ternary "golden" representation (77) relatively to 0-th digit, we find that the left part of any ternary "golden" representation (77) is mirror reflection of its right part, that is,

$$b_k = b_{-k}; b_{k-1} = b_{-(k-1)}; \ldots; b_2 = b_{-2}; b_1 = b_{-1}$$

This **property of the "mirror symmetry"** of the ternary "golden" representations (77) is fundamental property of the ternary "golden" numeral system (76). Table 13 demonstrates this property for some positive integers in the range {0,1,2,…,10}.

Thus, thanks to this simple investigation we have discovered the fundamental property of integers called *"mirror-symmetric property of integers*. Basing on this fundamental property, the "ternary "golden" numeral system," given by (76), is called *ternary "golden" mirror-symmetric numeral system* [29].

8.2.4 Conversion of numbers from positive to negative by using "ternary inversion"

Let us introduce a notion of *ternary inversion* (see Table 14).

Table 14. Ternary inversion

1	$\rightarrow$	$\bar{1}$
0	$\rightarrow$	0
$\bar{1}$	$\rightarrow$	1

The "ternary inversion" (Table 14) can be used for the conversion of the ternary "golden" representation of positive number N to the ternary "golden" representation of negative numbers ($-N$) and *vise versa.*

As it follows from Table 13, the positive number 9 has the following ternary "golden" representation:

$$9 = 011\bar{1},110\,, \tag{78}$$

what corresponds to the following sum:

$$\begin{aligned} 9 &= 0\times\Phi^6 + 1\times\Phi^4 + 1\times\Phi^2 + \bar{1}\times\Phi^0 + 1\times\Phi^{-2} + 1\times\Phi^{-4} + 0\times\Phi^{-6} \\ &= \left(\Phi^4 + \Phi^{-4}\right) - \Phi^0 + \left(\Phi^2 + \Phi^{-2}\right), \text{where } \Phi = \frac{1+\sqrt{5}}{2}. \end{aligned} \tag{79}$$

Since $\Phi^4 + \Phi^{-4} = 7, \Phi^2 + \Phi^{-2} = 3, \Phi^0 = 1$, then from (79) there follows that "golden" ternary representation (78) is true.

Example 6. Conversion of ternary "golden" representations of positive number 9 to negative number (-9)

n	3	2	1	0	-1	-2	-3
Φ^{2n}	Φ^6	Φ^4	Φ^2	Φ^0	Φ^{-2}	Φ^{-4}	Φ^{-6}
+9	0	1	1	$\bar{1}$,	1	1	0
↓	↓	↓	↓	↓	↓	↓	↓
-9	0	$\bar{1}$	$\bar{1}$	1,	$\bar{1}$	$\bar{1}$	0

8.2.5 The radix of the ternary "golden" mirror-symmetric numeral system

It follows from (76) that the radix of the numeral system (76) is the square of the golden ratio, that is,

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

This means that the numeral system (76) is a number system with an irrational radix.

The radix of the numeral system (76) has the following "golden" representation:

$$\Phi^2 = 10\,.$$

8.3 Mirror-symmetrical summation and subtraction

8.3.1 Mirror-symmetric summation

The following identities for the golden ratio powers underlie the mirror-symmetric summation:

$$2\,\Phi^{2k} = \Phi^{2(k+1)} - \Phi^{2k} + \Phi^{2(k-1)}\,; \tag{80}$$

$$3\,\Phi^{2k} = \Phi^{2(k+1)} + 0 + \Phi^{2(k-1)}\,; \tag{81}$$

$$4\,\Phi^{2k} = \Phi^{2(k+1)} + \Phi^{2k} + \Phi^{2(k-1)}\,, \tag{82}$$

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

The identity (80) is mathematical basis for the mirror-symmetric summation of two single-digit ternary digits and gives the rule of the carry-over formation (Table 15).

Table 15. Mirror-symmetric summation

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

The main peculiarity of Table 15 consists in the summation rule of two ternary units with equal signs, i.e.

$$\begin{array}{ccccccc} a_k & + & b_k & = & c_k & s_k & c_k \\ 1 & + & 1 & = & 1 & \bar{1} & 1 \\ \bar{1} & + & \bar{1} & = & \bar{1} & 1 & \mathbf{\bar{1}} \end{array}, \tag{83}$$

where a_k and b_k are the "treats" of the k-th digit.

We can see that for the mirror-symmetric summation of the treats of the one and same sign there arises the intermediate sum s_k with the opposite sign and the carry-over c_k with the same sign. However, the carry-over from the k-th digit spreads simultaneously to the next two digits, namely to the left-hand, that is, $(k+1)$-th digit, and to the right-hand, that is, $(k - 1)$-th digit.

Example 7. Summation of two ternary mirror-symmetric numbers 5 + 10:

$$\begin{array}{ccccccccc} 5 & = & 0 & 1 & \bar{1} & 1, & \bar{1} & 1 & 0 \\ 10 & = & 0 & 1 & 1 & 0, & 1 & 1 & 0 \\ S_1 & = & 0 & \bar{1} & 0 & 1, & 0 & \bar{1} & 0 \\ C_1 & & 1 & \leftrightarrow & 1 & & 1 & \leftrightarrow & 1 \\ \hline 15 & = & 1 & \bar{1} & 1 & 1, & 1 & \bar{1} & 1 \end{array}$$

Notice that the symbol $\leftrightarrow$ marks the process of carry spreading.

We can see that the summation process for this example consists of two steps. The first step is forming the first multi-digit intermediate sum S_1 and the first multi-digit carry-over C_1 according to Table 15. The second step is summation of the numbers $S_1 + C_1$ according to Table 15. Because for this case the second multi-digit intermediate carry-over $C_1 = 0$, this means that the ternary mirror-symmetric summation is over and the sum $S_1 + C_1 = 15$ is the result of mirror-symmetrical summation. It is important to emphasize that the result of mirror-symmetrical summation:

$$15 = 1\ \bar{1}\ 1\ 1\ 1,\ 1\ \bar{1}\ 1 \tag{84}$$

is represented in the mirror-symmetrical form.

8.3.2 Mirror-symmetric subtraction

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

$$N_1 - N_2 = N_1 + (- N_2). \tag{85}$$

We can point on a number of the important advantages of the mirror-symmetrical summation and subtraction from the engineering point of view:

(1) The mirror-symmetric subtraction is reduced to the mirror-symmetrical summation by the use of the rule (85).
(2) The mirror-symmetrical summation and subtraction is performed in the "direct" code, that is, without the use of the notions of the "inverse" and "additional" codes.
(3) The sign of the summarized numbers is defined automatically because it coincides with the sign of the higher significant ternary numeral of the ternary mirror-symmetrical representation of the summation result.
(4) The summation result is represented always in the mirror-symmetrical form that allows checking a process of the ternary mirror-symmetric summation and subtraction.

8.4 Mirror-symmetrical multiplication and division

8.4.1 Mirror-symmetrical multiplication

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

$$\Phi^{2n} \times \Phi^{2m} = \Phi^{2(n+m)} . \tag{86}$$

The rule of the mirror-symmetrical multiplication of two single-digit ternary mirror-symmetrical numbers is given in Table 16.

Table 16. Mirror-symmetrical multiplication

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

The ternary mirror-symmetrical multiplication is performed in the "direct" code. The general algorithm of the multiplication of two multi-digit mirror-symmetrical numbers is reduced to the formation of the partial products in accordance with Table 16 and their summation in accordance with the rule of the mirror-symmetrical summation.

Example 8. Multiply the negative mirror-symmetric number - 6 = $\bar{1}$ 0 1, 0 $\bar{1}$ by the positive mirror-symmetric number 2 = 1 $\bar{1}$, 1:

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

The multiplication result in the example 8 is formed as the sum of the three partial products. The first partial product $\bar{1}$ 0, 1 0 $\bar{1}$ is the result of multiplication of the mirror-symmetrical multiplier - 6 = $\bar{1}$ 0 1, 0 $\bar{1}$ by the lowest positive unit of the mirror-symmetrical multiplier 2 =1 $\bar{1}$,1, the second partial product 1 0 $\bar{1}$, 0 1 is the result of the multiplication of the same number - 6 = $\bar{1}\ 0\ 1,\ 0\ \bar{1}$ by the middle negative unit of the number 2=1 $\bar{1}$, 1, and, finely, the third partial product $\bar{1}\ 0\ 1,\ 0\ \bar{1}$ is the result of the multiplication of the same number - 6 = $\bar{1}\ 0\ 1,\ 0\ \bar{1}$ by the higher positive unit of the number 2 = 1 $\bar{1}$, 1.

Notice that the product -12 = $\bar{1}\ 1\ 0\ \bar{1},\ 0\ \bar{1}\ \bar{1}$ is represented in the mirror-symmetric form! Because its higher digit is a negative unit $\bar{1}$, it follows from here that the product is a negative mirror-symmetrical number.

8.4.2 Mirror-symmetrical division

The ternary mirror-symmetrical division is performed in accordance with the division rule of the classical ternary symmetrical numeral system [48]. The general algorithm of the ternary mirror-symmetrical division is reduced to the sequential subtraction of the shifted divisor, which is multiplied by the next ternary numeral of the quotient.

8.4.3 A concept of the ternary pipelined mirror-symmetrical adder

The detailed description of this concept is given in [29]. This concept is of great interest for pipelined signal microprocessors, where high performance is combined with high checking of data processing, based on the "principle of mirror symmetry."

8.4.4 The main arithmetical advantages of the mirror-symmetrical multiplication and division

We can formulate the following arithmetical advantages of the mirror-symmetrical multiplication and division:

(1) The mirror-symmetric multiplication and division is reduced to the mirror-symmetrical summation
(2) The mirror-symmetrical multiplication and division are performed in the “direct” code, that is, without the use of the notions of the “inverse” and “additional” codes
(3) The sign of the results of the mirror-symmetrical multiplication and division is defined automatically because it coincides with the sign of the higher significant ternary numeral of the ternary mirror-symmetrical representation of the result of the mirror-symmetrical multiplication and division
(4) The results of the mirror-symmetrical multiplication and division are represented always in the mirror-symmetrical form that allows checking a process of the ternary mirror-symmetrical multiplication and division
(5) The ternary “golden” mirror-symmetrical arithmetic retains all advantages of the classical ternary numeral system, but its main advantage is a possibility of checking all arithmetic operations according to the principle of "mirror symmetry;" this creates real prerequisites for using the ternary "golden" mirror-symmetrical numeral system for the design of highly reliable computers for mission-critical applications.

9 “Golden” Self-correcting Digital Metrology

9.1 The “binary” resistive divisor

In engineering practice the so-called *resistive divisors*, intended for the division of electric currents and voltages in the given ratio, are widely used. One of the variants of such divisor is shown in Fig. 6.

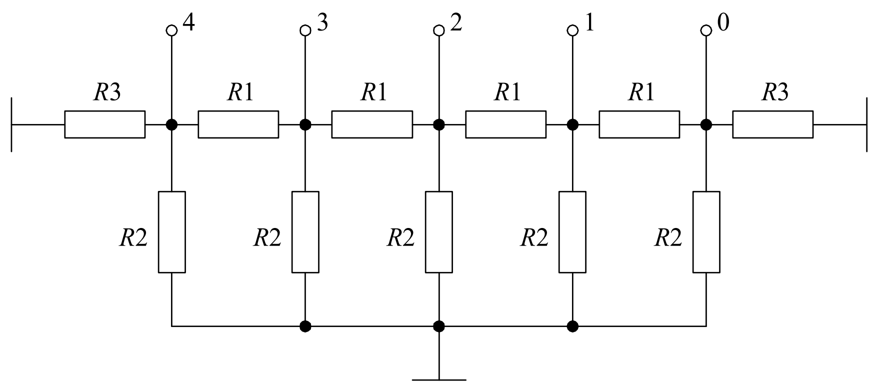


Fig. 6. The "binary" resistive divisor

The "binary" resistive divisor in Fig. 6 consists of the "horizontal" resistors of the kind $R1$ and $R3$ and the "vertical" resistors $R2$. The resistors of the divisor are connected between themselves by the "connecting points" 0, 1, 2, 3, 4. Each point connects three resistors, which form together the resistive section. Notice that Fig. 6 shows the resistive divisor, which consists of the 5 resistive sections

First of all, we notice that the parallel connection of the resistors $R2$ and $R3$ to the right of the "connecting point" 0 and to the left of the "connecting point" 4 can be replaced by the equivalent resistor with the resistance, which can be calculated according to the law on the resistor parallel connection:

$$R_{e1} = \frac{R2 \times R3}{R2 + R3} \tag{87}$$

Taking into consideration (87), it is easy to find the equivalent resistance of the resistive section to the right of the "connecting point" 1 and to the left of the "connecting point" 3:

$$R_{e2} = R_1 + R_{e1} \tag{88}$$

In dependence on the choice of the resistance values of the resistors $R1$, $R2$, $R3$ we can realize the different coefficients of the current or voltage division. Consider now the so-called "binary" divisor, which consists of the following resistors: $R1=R$; $R2=R3=2R$, where R is some standard resistance value. For this case the expressions (87), (88) take the following values:

$$R_{e1} = R; \qquad R_{e2} = 2R. \tag{89}$$

Then, taking into consideration (88), we can find that the equivalent resistance of the resistor circuit to the left or to the right of any "connecting point" 0, 1, 2, 3, 4 is equal to $2R$. This means that the equivalent resistance of the divisor in the "connecting points" 0, 1, 2, 3, 4 can be calculated as the resistance of the parallel connection of the three resistors of the value $2R$. Using the electrical circuit laws we can calculate the equivalent resistance of the divisor in each "connecting point" 0, 1, 2, 3, 4:

$$R_{e3} = \frac{2}{3}R\,. \tag{90}$$

Connect now the generator of electric current I to one of the "connecting points", for example, to the point 2. Then according to Ohm's law the following electric voltage will appear in this point:

$$U = \frac{2}{3} RI . \tag{91}$$

Find now the electrical voltages in the "connecting points" 3 and 1, which are adjacent to the point 2. It is easy to show that the voltage transmission coefficient between the adjacent "connecting points" is equal to $\frac{1}{2}$. This means that the "binary" divisor fits very well to the binary system and this fact is a cause of wide use of the "binary" divisor in modern digital metrology, including digit-to-analog and analog-to-digit converters.

9.2 The "golden" resistive divisors

Choose the values of the resistors in Fig. 6 as follows:

$$R1 = \Phi_p^{-p} R;\ R2 = \Phi_p^{p+1} R;\ R3 = \Phi_p R , \tag{92}$$

where Φ_p is the golden p-proportion, $p \in \{0, 1, 2, 3, \ldots\}$.

We note that in its electrical structure the "golden" resistive divisor coincides with the "binary" resistive divisor in Fig. 6. It is clear that the "golden" resistive divisor in Fig. 6 gives an infinite number of the different resistive divisors because every p "generates" a new divisor. In particular, for the case $p = 0$ the value of the golden (p=0) proportion Φ_0=2 and the "golden" divisor is reduced to the classical "binary" divisor of the kind $R - 2R$.

For the case p = 1 the resistors $R1$, $R2$, $R3$ take the following values:

$$R1 = \Phi^{-1} R;\ R2 = \Phi^2 R;\ R3 = \Phi R , \tag{93}$$

where $\Phi = \frac{1+\sqrt{5}}{2}$ is the golden proportion.

Taking into consideration the usage of the golden p-proportions Φ_p in the resistors (92,93), we will name the resistive divisors, given by (92) and (93), the *"golden" resistive divisors.*

Taking into consideration (92) and using the following mathematical identities for the *golden p-proportions*:

$$\Phi_p = 1 + \Phi_p^{-p} , \tag{94}$$

$$\Phi_p^{p+2} = \Phi_p^{p+1} + \Phi_p , \tag{95}$$

the following values of the equivalent resistance of the resistive circuit of the "golden" resistive divisor in Fig. 6 to the left and to the right from the "connecting points" 0 and 4 can be obtained [6,31]:

$$R_{e1} = \frac{R2 \times R3}{R2 + R3} = \frac{\Phi_p^{p+1} R \times \Phi_p R}{\Phi_p^{p+1} R + \Phi_p R} = R . \tag{96}$$

Using (88) and (94), it is easy to prove that the equivalent resistance R_{e2} is equal:

$$R_{e2} = \Phi_p^{-p} R + R = \Phi_p R . \tag{97}$$

By using the relations (94), (95), it is proved in [6] that the voltage transmission coefficient between the adjacent "connecting points" of the "golden" resistor divisor, given by (92), is equal to the reciprocal to the golden p-proportion Φ_p !

Thus, the "golden" resistive divisor in Fig. 6, based on the golden p-proportions Φ_p, are quite real electrical circuits. It is clear that the theory of the "golden" resistive divisors, described in the article [6] and the book [31], is new source for the development of the "digital metrology" and analog-to-digit and digit-to-analog converters.

9.3 The use of the codes of the golden *p*-proportions in digit-to-analog and analog-to-digit converters

9.3.1 The "golden" digit-to-analog converters

The electrical circuit of the «golden» digit-to-analog converter (DAC), based on the «golden» resistive divisor in Fig. 6, is shown in Fig. 7.

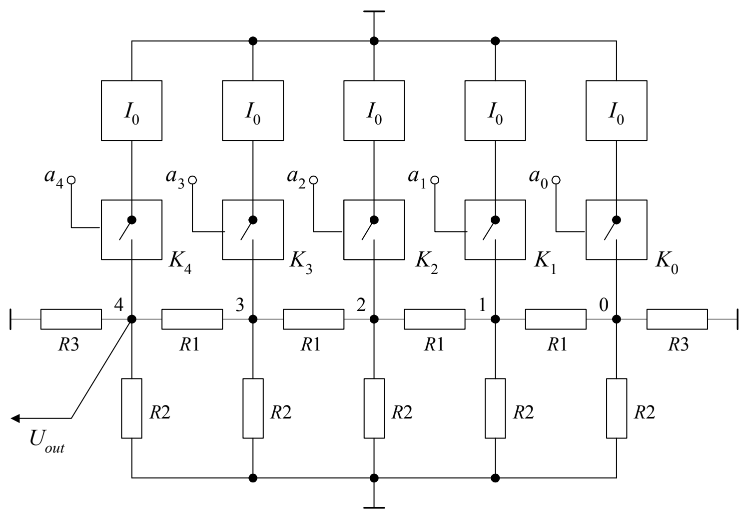


Fig. 7. The «golden» DAC

Note that the «golden» DAC in Fig. 7 includes the 5 digits. However the number of the DAC digits may be increased to some arbitrary n by extending the "golden" resistive divisor to the left and to the right.

The «golden» DAC contains the 5 (n in the general case) generators of the standard electrical current I_0 and the 5 (n in general case) electrical current keys K_0 - K_4. The key states are controlled by the binary digits of the "golden" code word $a_4 a_3 a_2 a_1 a_0$. For the case a_i = 1 the key K_i is closed, for the case a_i = 0 is open (i = 0, 1, 2, ..., n).

One can show [6,31] that the closed key K_i results in the following voltage in the i-th point of the resistive divisor:

$$U_i = \beta_p I_0 R,$$

where

$$\beta_p = \frac{1}{1+\Phi_p^{-1}}.$$

As the voltage potential U_i is passed from the i-th point to the (i+1)-th point with the transmission coefficient $\frac{1}{\Phi_p}$, the following voltage value appears at the DAC output:

$$U_{out} = \frac{\beta_p I_0 R}{\Phi_p^{n-i-1}} \times \Phi_p^i.$$

Using the superposition principle, it is easy to show that the input code word of the code of the golden p-proportion $a_{n-1}a_{n-2}\ \dots\ a_0$ results in the following output voltage U_{out}:

$$U = B_p \sum_{i=0}^{n-1} a_i \Phi_p^i. \qquad (98)$$

where

$$B_p = \frac{\beta_p I_0 R}{\Phi_p^{n-1}}.$$

It follows from (98) that the electrical circuit in the Fig. 7 converts the “golden” code word $a_4 a_3 a_2 a_1 a_0$ into the electrical voltage U_{out} with the constant coefficient B_p.

9.3.2 Checking the «golden» DAC

In the measurement practice there is a necessity to check up the DAC linearity in the production and operation process. For the classical binary DAC the following correlation for the checking up the DAC linearity is used:

$$2^n = \sum_{i=0}^{n-1} 2^i + 1.$$

The mathematical properties of the code of the golden p-proportion (67) provide very wide possibility for checking up the DAC linearity. In particular, the checking up of linearity of the «golden» DAC, based on the classical golden ratio, is reduced to the checking up the following relations:

$$\Phi^n = \Phi^{n-1} + \Phi^{n-2} = \Phi^{n-1} + \Phi^{n-3} + \Phi^{n-4} = \Phi^{n-1} + \Phi^{n-3} + \Phi^{n-5} + \Phi^{n-6} = \dots \qquad (99)$$

Checking up linearity is performed in the following manner. We must check up that the output voltage of the «golden» DAC in Fig. 7 doesn't change for the following input code combinations:

1 0 0 0 0 0 0
0 1 1 0 0 0 0
0 1 **0 1 1** 0 0
0 1 0 1 **0 1 1**

Note that the different input code combinations are formed from the initial code combination 1000000 by means of the "devolutions".

9.3.3 Self-correcting «golden» ADC and DAC

The functioning algorithm and structural scheme of the «golden» ADC and DAC do not differ from the functioning algorithm and the structural scheme of the classical binary ADC and DAC. However, the properties of the codes of the golden *p*-proportions (67), in particular, the **property of plurality** of the "golden" representations of one and the same real number leads to interesting technical advantages of the "golden" ADC and DAC. In particular, the **property of plurality** allows designing the self-correcting "golden" ADC and DAC.

The detailed description of the self-correcting "golden" DAC is given in the works [6,31], the engineering developments of the self-correcting "golden" DAC and ADC is given in the brochure [13].

As it is well known, there is the notion of the metrological stability for the evaluation of the reliability of measurement systems and devices, in particular ADC and DAC. In ADC and DAC, the deviations of the parameters of analog elements of ADC and DAC (in particular, resistive divisors) from their standard values are the main cause of the non-stability of measurement systems. These deviations are caused by different interior and exterior factors ("aging" of analog elements, temperature influences, etc.) and they are usually the "slow" time functions. At the designing the measurement systems of high accuracy there is a problem decreasing the requirements to the technological exactness of the analog elements production and eliminating such difficult technological operations as the laser "trimming" of the analog elements (in particular resistive divisors).The solution of this problem is realized by application of **self-correcting principle**. The essence of this principle is as follows. We will introduce into the structure of ADC and DAC special checking device, comprising a stable electric element (for example, Zener diode). This device periodically determines the most typical errors ADC and DAC, associated with the drift of the parameters of analog elements.

Typical errors are: *"zero drift"*, *changing the slope of conversion characteristics*, and *violation of linearity* of the resistive divisor.

Here, correction of non-linearity of resistive divisor is the most difficult. For the first time, the theoretical substantiation of the effective correction of non-linearity of the resistive divisor, based on the codes of the golden *p*-proportions, was described in the 1978-article [6]. This led to the creation in 80th years of 20 century the "golden" self-correcting ADC [13] (Fig. 8).

Fig. 8. The "golden" self-correcting 18-digit ADC

The “golden” self-correcting ADC in Fig. 8 has the following specifications:

1. The number of bits - 18 (17 digital and one signed)
2. Time of conversion - 15 micro-sec
3. Total error - 0.006%
4. Linearity error - 0.003%
5. Frequency range - 25 kHz
6. Operating temperature range - 20±30°C

The ADC in Fig. 8 was designed on discrete electronic components. The ADC consisted of two functional devices:

1) The device for metrological checking of “zero drift”, changing the slope of conversion characteristics, and violation of linearity of the resistive divider
2) The device for analog-to-digit conversion based on the “golden” resistive divisor of the kind (93)

The ADC in Fig. 8 operates in two modes:

1) The mode of metrological checking
2) The mode of analog-to-digit conversion

Due of the device for metrological checking, the ADC has the following useful properties:

1. ADC production process does not require laser trimming of the resistive divisor. Setting the required accuracy is done automatically through the use of the device for metrological checking. For example, technological error of analog components in ADC in Fig. 8 declined in 1000 times in comparison with the required accuracy of analog-to-digital conversion. This reduces production cost of ADC.
2. Metrological parameters of ADC are not dependent from changes of temperature and aging analog elements, such as resistive divider.

Thus, the main conclusion is the fact that the "golden" ADC in Fig. 8 is "a dream" of producer on designing measuring devices with the "eternal" metrological parameters, which are not dependent from technological errors, changes of temperature and aging analog elements.

The “golden” ADC in Fig. 8 was recognized in Soviet Union as the best ADC and it was used widely by the Soviet leading metrological firms. ADC was awarded the "Gold Medal" of Exhibition of Economic Achievements of the USSR (Moscow).

Note that in the Vinnitsa Technical University (Ukraine) the self-correcting DAC with high technical specifications was designed [13].

9.3.4 A concept of the “golden” analog microprocessor

If we take into consideration the fact that the ADC and DAC are important input and output devices for microprocessors and microcontrollers for mission-critical applications (in particular, space system), then we can conclude that the task of designing of highly reliable microprocessors and microcontrollers, based on the "golden" ADC and DAC with "eternal" metrological specifications and Fibonacci and “golden" microprocessors of highest noise immunity is one of the most urgent problems of modern digital microelectronics and digital metrology.

In this connection we should pay special attention to **analog microprocessors.** In the last decades the concept of analog signal microprocessors got wide spreading. The promise of this trend is emphasized in the title of the article **“Analog microprocessors could be the unforeseen future of computing”** [49], presented by the DARPA on the Internet. It is clear that the above “golden” self-correcting ADC and DAC and "golden" arithmetic are of great interest for this latest trend of microelectronic technology, as it can lead to the creation of the "golden" analog microchip with high metrological stability and noise immunity what is very important for many mission-critical applications.

10 Conclusion

10.1 Numeral systems in their historical development

In the process of historical development, starting from Babylonian and Egyptian mathematics up present time, an attitude to numeral systems had changed. During the last centuries the decimal system has been introduced widely into education and computing practise, besides, during the last decades the binary system became one of central notion of computer science. It seemed that the decimal and binary numeral systems do not have alternatives.

Unfortunately, study of numeral systems was far from problems of contemporary "pure" mathematics and therefore in this part mathematics not progressed far in comparison to the period of its origin.

10.2 An interest in numeral systems in modern computer science and technology

However, the rapid development of computer science and technology has led to changing the attitude to numeral systems. A problem of numeral systems became one of the actual problems of modern computer science. In this area a huge interest in methods of number representation and new computer arithmetic's was again arisen. During second half of 20 century, the numeral systems with the "exotic" titles and properties appeared: *system for residual classes*, *ternary symmetrical numeral system*, *numeral system with the complex radix, nega-positional, factorial, binomial numeral systems* [50,51], and also *Bergman's system* [48], *Fibonacci p-codes and codes of the golden p-proportions* [14,15]. All of them had those or other advantages in comparison with the binary system and were pointed on the improvement of those or other computer characteristics; some of them became a basis for the creation of new computer projects and conceptions (the ternary computer "Setun", processors based on system for residual classes, Fibonacci and "golden" computers and so on).

But there is also other interesting aspect of this problem. Later 4 millennia after the invention by Babylonians of the *positional principle* of number representation, we can look a peculiar "Renaissance" in the field of numeral systems [14,15,50,51]. Due to the efforts first of all of the experts in computer science, mathematics as though again returned back to the period of its origin, when the numeral systems had defined a topic and essence of all mathematics (Babylon, Ancient Egypt, India and so on).

But then we can put the following question: possibly the modern numeral systems, created for computer needs, can influence on the development of number theory and by such way could influence not only on the development of computer science, but also of all mathematics. A search of the answer to this question is one of the major goals of the present article, devoted to new class of positional numeral systems, *Fibonacci and "golden" numeral systems.*

10.3 The main stages in the development of Fibonacci and "Golden" positional numeral systems

We can specify the following stages in the development of the Fibonacci and "Golden" positional numeral systems:

10.3.1 Bergman's system

In 1957 the young American mathematician George Bergman made important mathematical discovery in the field of numeral systems. For the first time, he introduced into being a *numeral system with an irrational* *base* [48]. Possibly, Bergman's system [48] is the most important mathematical discovery in the field of numeral systems after the discovery of positional principle of number representation (Babylon, 2000 B.C.) and decimal system (India, 6-8th centuries). Although Bergman's system [48] was the result of a principal importance for the numeral systems theory, however in that period Bergman's system simply wasn't noted neither by mathematicians nor by engineers.

10.3.2 Fibonacci *p*-codes

In 70'th years of 20th century, the author of present article developed the so-called *Fibonacci p-codes* [4-5]. These new positional numeral systems followed from *algorithmic measurement theory* [14] and led to new conception of computers, *Fibonacci computers* as an alternative to classical binary computers.

10.3.3 Codes of the golden *p*-proportions

In 80'th years of 20th century the author of the present article has introduced a new class of the numeral systems with irrational bases, the *codes of the golden p-proportion* [6-8]. These numeral systems were a wide generalization of the classical "binary system" (p=0) and Bergman's system (p=1) [48]. The "golden" computer arithmetic follows from Bergman's system [48] and codes of the golden p-proportions [6-8, 15].

10.3.4 Ternary mirror-symmetrical arithmetic

In 2002 "The Computer Journal" (British Computer Society) has published author's article "*Brousentsov's ternary principle, Bergman's number system and ternary mirror-symmetrical arithmetic"* [29]. This ternary numeral system is original unification of Bergman's system [45] and Brousentsov's ternary principle and is fundamentally new ternary numeral system with irrational base $\Phi^2 = \frac{3+\sqrt{5}}{2} \approx 2.618$. This new ternary numeral system has unique property of *mirror symmetry,* which can be used for checking all arithmetical operations. The prominent American mathematician and a world-renowned expert in computer science **Donald Ervin Knuth** was the first outstanding scientist who congratulated the author with the publication of the article [29].

10.3.5 Applications of the Fibonacci and "Golden" positional numeral systems

Fibonacci p-codes, Bergman's system [48] and codes of the golden p-proportions [6-8,15] have a number of very important applications. On the first hand, they are of great number-theoretical importance for number theory, because they can be considered as a new constructive definition of natural and real numbers [34]. The new unexpected properties of natural numbers (Z-property, F- and L-codes), following from this definition [34], is a brilliant confirmation of fundamental importance of such approach to number theory. On the other hand, Fibonacci p-codes [4-6,14], Bergman system [48] and its generalization, codes of the golden p-proportions [6-8,15], have utmost importance for the development of modern computer science and digital metrology for mission-critical applications (the "golden" self-correcting DAC and ADC).

10.3.6 The main purpose of the Fibonacci and "Golden" positional numeral systems

The main purpose of the Fibonacci and "Golden" positional numeral systems is to unite mathematics, computer science and digital metrology into a whole unit, as it was in ancient Egypt and Babylon, when mathematics begun to develop. The main goal of this union is to design high-reliable computing and measuring systems of future with unique technical specifications for mission-critical applications. These new positional numeral systems can lead to a new stage in the development of mathematics, computer technology and digital metrology.

Competing Interests

Author has declared that no competing interests exist.

References

[1] Von Neumann architecture. From Wikipedia, the free encyclopaedia. Available:https://en.wikipedia.org/wiki/Von_Neumann_architecture

[2] Hetagurov JA. Ensuring the national security of real-time systems. BC / NW. 2009;2(15):11.1 (Russian).

[3] Yanushkevich Svetlana N, Lyshevski Sergey E, Shmerko Vlad P. Computer arithmetics for nanoelectronics. CRC PRESS: Boca Raton, Ann Arbor, London, Tokyo.

[4] Stakhov AP. Redundant binary positional number systems. In the book "Homogenous digital computer and integrated structures." Taganrog: Publishing House "Taganrog Radio University". 1974;2:3-40 (Russian).

[5] Stakhov AP. A use of natural redundancy of the Fibonacci number systems for computer systems control. Automation and Computer Systems. 1975;6:80-87 (Russian).

[6] Stakhov AP. Digital metrology on the base of the Fibonacci codes and golden proportion codes. In the book “Contemporary Problems of Metrology.” Moscow: Publishing House of Moscow Machine-building Institute. 1978;51-65 (Russian).

[7] Stakhov AP. Fibonacci and “Golden” ratio codes. In the book “Fault-tolerant Systems and Diagnostic FTSD-78,” Gdansk. 1978;276-285.

[8] Stakhov AP. The golden mean in the digital technology. Automation and Computer Systems. 1980;1:27-33 (Russian).

[9] Stakhov AP. Algorithmic measurement theory and fundamentals of computer arithmetic. Measurement. Control. Automation. 1988;2:3-21 (Russian).

[10] Stakhov AP. The golden section in the measurement theory. Computers & Mathematics with Applications. 1989;17(4-6):613-638.

[11] Stakhov AP. The golden proportion principle: Perspective way of computer progress. Bulletin of the National Academy of Sciences of Ukraine. 1990;1-2:31-48 (Ukrainian).

[12] Stakhov AP. Algorithmic measurement theory: A general approach to number systems and computer arithmetic. Control Systems and Computers. 1994;4-5:10-31 (Russian).

[13] Noise-tolerant codes. Fibonacci computer. Moscow: Publishing House “Nauka”; 1989 (Russian).

[14] Stakhov AP. Introduction into algorithmic measurement theory. Moscow: Publishing House “Soviet Radio”; 1977 (Russian).

[15] Stakhov AP. Codes of the golden proportion. Moscow: Publishing House “Radio and Communication; 1984 (Russian).

[16] Reduction method of p-Fibonacci code to the minimal form and device for its realization. Patent certificate of USA No 4187500.

[17] Device for reduction of p-Fibonacci codes to the minimal form. Patent certificate of USA No 4290051.

[18] Reduction method of *p*-Fibonacci code to the minimal form and device for its realization. Patent certificate of England No 1543302.

[19] Device for reduction of *p*-Fibonacci codes to the minimal form. Patent certificate of England No 2050011.

[20] Reduction method of *p*-Fibonacci code to the minimal form and device for its realization. Patent certificate of Germany No 2732008.

[21] Device for reduction of *p*-Fibonacci codes to the minimal form. Patent certificate of Germany No 2921053.

[22] Reduction method of *p*-Fibonacci code to the minimal form and device for its realization. Patent certificate of Japan No 1118407.

[23] Reduction method of *p*-Fibonacci code to the minimal form and device for its realization. Patent certificates of France No 7722036, No 2359460.

[24] Device for reduction of *p*-Fibonacci codes to the minimal form. Patent certificates of France No 7917216, No 2460367.

[25] Reduction method of *p*-Fibonacci code to the minimal form and device for its realization. Patent certificate of Canada No 1134510.

[26] Device for reduction of *p*-Fibonacci codes to the minimal form. Patent certificate of Canada N1132263.

[27] Reduction method of *p*-Fibonacci code to the minimal form and device for its realization. Patent certificate of Poland No 108086.

[28] Reduction method of *p*-Fibonacci code to the minimal form and device for its realization. Patent certificate of DDR No 150514.

[29] Stakhov AP. Brousentsov's ternary principle, Bergman's number system and ternary mirror-symmetrical arithmetic. The Computer Journal. 2002;45(2):222-236.

[30] Stakhov A. Fibonacci matrices, a generalization of the "Cassini formula", and a new coding theory. Chaos, Solitons & Fractals. 2006;30(1):56-66.

[31] Stakhov AP. The mathematics of harmony. From Euclid to Contemporary Mathematics and Computer Science. Assisted by Scott Olsen. New Jersey, London, Singapure, Beijing, Shanghay, Hong Kong, Taipei, Chennai: World Scientific; 2009.

[32] Borisenko AA, Stakhov AP. On Fibonacci counters. Bulletin of the Sumy State University (Ukraine). 2011;3:141-149.

[33] Stakhov A, Borisenko A, Matsenko S. Fibonacci counter based on Zeckendorf's theorem (Boolean Realization). Visual Mathematics. 2014;16(2). Available:http://elib.mi.sanu.ac.rs/files/journals/vm/57/vmn57p1-19.pdf

[34] Stakhov AP. The "Golden" number theory and new properties of natural numbers. British Journal of Mathematics and Computer Science. 2015;11(6):1–15.

[35] Monteiro P, Newcomb RW. (University of Maryland). Minimal and maximal Fibonacci representations: Boolean generation. The Fibonacci Quarterly. 1976;14(1).

[36] Licomendes P, Newcomb R. (University of Maryland). Multilevel Fibonacci conversion and addition. The Fibonacci Quarterly. 1984;22(3).

[37] Ligomenides P, Newcomb R. Equivalence of some binary, ternary and quaternary Fibonacci computers. Proceeding of the Eleventh International Symposium on Muliple-Valued Logik, Norman, Oklahoma; 1981.

[38] Ligomenides P, Newcomb R. Complement representations in the Fibonacci computer. Proceedings of the Fifth Symposium on Computer Arithmetic, Ann Arbor, Michigan; 1981.

[39] Newcomb R. Fibonacci numbers as a computer base. Conference Proceedings of the Second Interamerican Conference on Systems and Informatics, Mexico City; 1974.

[40] Hoang VD. A class of arithmetic burst-error-correcting codes for the Fibonacci computer. PhD Dissertation, University Maryland; 1979.

[41] George Polya. Mathematical discovery. Publishing House "John Wiley & Sons"; 1962.

[42] Vorobyov NN. Fibonacci numbers. Third edition. Moscow: Nauka, (the first edition, 1961) (Russian); 1969.

[43] Hoggatt VE Jr. Fibonacci and Lucas numbers. Boston, MA: Houghton Mifflin; 1969.

[44] Vajda S. Fibonacci & Lucas numbers and the golden section. Theory and Applications. Ellis Harwood Limited; 1989.

[45] Kharkevitch AA. A struggle against noises. Moscow: Publishing House "Fizmatgiz"; 1963 (Russian).

[46] Ancient Egyptian multiplication. From Wikipedia, the free encyclopaedia. Available:https://en.wikipedia.org/wiki/Ancient_Egyptian_multiplication

[47] Kautz WH. Fibonacci codes for synchronization control. IEEE Trans. Inform. Theory. 1965;11(8): 284-292.

[48] Bergman GA. A number system with an irrational base. Mathematics Magazine. 1957;31:98-119.

[49] DARPA UPSIDE: Analog microprocessors could be the unforeseen future of computing. Available:https://www.patexia.com/feed/darpa-upside-analog-vs-digital-microprocessors-20120917

[50] Pospelov DA. Arithmetical foundations of computers. Moscow: Publishing House "Vyshaja shkola"; 1970 (Russian).

[51] Kasatkin VN. News about numeral systems. Kiev: Publishing House "Vyscha shkola"; 1989 (Russian).

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/14921