*i*Journals: International Journal of Software & Hardware Research in Engineering
ISSN-2347-4890
Volume 8 Issue 2 February 2020

Computer Arithmetic of Numbers, Vectors, Figures and Functions. Algorithms and Hardware

Author: Solomon I. Khmelnik [1]

E-mail: solik@netvision.net.il

ABSTRACT

The article contains a prospectus of the book under the same title [1]. This book is published only in Russian and in this connection, this prospectus is published. The book contains 673 pages. The author seeks assistance in publishing a book in English.

Keywords: Computer Arithmetic, Numbers, Vectors, Figures, Functions, Algorithms, Hardware

1. ANNOTATION TO BOOK

The book describes the author's proposed little-known methods of coding mathematical objects - real and complex numbers, multidimensional vectors, flat and spatial geometric shapes, functions of one and many arguments. The properties of received codes, algorithms of various operations with them, hardware implementation of these algorithms with the purpose of construction of specialized processors are considered. The theory is illustrated by numerous examples.

2. INTRODUCTION IN BOOK

Universality and high efficiency of modern computers frequently creates representation about completeness of the principles their constructions and functioning. The life, however, denies such representation, showing the problems taking place on a limit of computer opportunities at productivity and demanding excessive efforts at programming. In reply to it is offered to create multiprocessing systems, the matrix processors, homogeneous structures and, in general, that or a diverse way to carry out computing operations with array of numbers. Addition and frequently alternative to these ways is development of the computer arithmetic unit processing complex mathematical objects which in usual computers are represented by set of real numbers. Quick-action of such arithmetic units essentially grows, if complex mathematical objects manage to be presented a uniform binary code, with which computing operations specific to these objects are feasible. The arithmetic unit for operations with codes of complex numbers below is described.

The following describes such arithmetic devices.

Let us briefly discuss the history of the issue. Computer arithmetic of complex mathematical objects originates in the article by Shannon on the positional coding of real numbers on a negative basis [1]. This idea, apparently, was first implemented in Poland [2] and prompted (apparently) several authors to develop methods for coding complex numbers. Almost simultaneously, Knut [3] proposed a coding system for the base j√2. Khmelnik [4] proposed several systems, including on the bases of j√2 and (-1+j). The base (-1+j) was later reviewed by Penney [5]. Khmelnik in his dissertation [6] considered a set of issues for constructing an arithmetic unit for operations with complex numbers. These results were further developed in [7, 8, 9, 11, 12, 13, 14, 33, 34, 35, 44].

In several papers [16, 17, 18], methods for constructing multipliers of complex numbers are considered. The focus is on how to implement these devices on a chip. For this, redundant coding systems are proposed, which, according to the authors, allow us to construct more regular schemes. However, this does not consider other operations with the proposed codes (for example, division).

For codes of real numbers, the method ‘digit by digit‘ [19, 20] for hardware calculation of elementary functions is known. It can be generalized to positional codes of complex numbers, which was first done by Khmelnik in [6, 11]. Moreover, it is often sufficient to have a hardware implementation of only potentiations and logarithms, since through these functions in the complex domain all elementary functions can be expressed. In addition, this method is applicable to the construction of algorithms for the hardware solution of transcendental equations and systems of such equations. When using codes of complex (and not real) numbers, the class of such equations is expanded, and the algorithms for solving them are greatly simplified. In [6, 11] one of such algorithms is described.

Further development of the idea of positional coding followed the path of constructing positional codes of vectors [21, 22], matrices [36, 37], functions [23, 24, 25, 33, 47], geometric figures [22, 26, 27, 28, 32, 38, 39, 46]. It should be noted that codes of geometric figures can be considered as codes of numerical arrays and effective search algorithms can be constructed for them [29, 30, 31]. Many of these results are summarized in the book [32].

The preference given to positional codes is explained mainly by the fact that arithmetic operations are very simple with them. So, regardless of the object of coding, the addition of positional codes is associated with hyphenation from lower-order digits to higher-order ones, and multiplication consists of shifts (that is, the renumbering of digits) and additions. The above-mentioned method 'digit by digit is generally applicable only in combination with the positional coding system.

It is important to note that for the programming of the computers uses the existing mathematical apparatus, which naturally does not take into account the specific capabilities of computers. It is hoped that the distribution of proposed computers will not only find other methods for solving problems but also other unexpected applications, as it continuously happens with existing computers. For example, there is a theory of functions of a spatial complex variable [40]. The algebra of four-dimensional vectors [21, 32], proposed for their coding, coincides with the algebra of spatial complex numbers used in [40]. In this regard, it becomes possible to develop computer arithmetic of spatial complex numbers with hardware calculating the functions of this variable, as a further generalization of the method 'digit by a digit (just as it was done for complex numbers [6, 11]). This has a practical meaning since the theory of functions of a spatial complex variable can be used in very complex problems of theoretical physics [40].

In this book, you can find a lot of analogies with traditional computer arithmetic. We can specify the number of books where this arithmetic is described in detail [41, 42, 43].

This book includes books [32, 45-48, 51, 52] and is their continuation. In [53, 58] the project of an arithmetic unit is described, in which many results of the proposed theory are embodied. This project is a detailed description of the VHDL model of an arithmetic unit that performs about 400 different operations with binary codes of real numbers on a negative basis and with binary codes of complex numbers.

The book is focused on the user who intends to apply the described computer arithmetic in their own developments of specialized processors. To this end, the book includes all the information necessary to

- understanding of the functioning of the processor in detail;
- use of the technical solutions given in the book for own development.

In other words, the book includes

- Coding Theory,
- Algorithms of operations,
- Examples of encoding, decoding, and computing,
- Description of several processor options,
- Command systems for them,
- Operational block diagrams,
- Comparative analysis.

3. SHORT CONTENT

Foreword
References
Part 1. Codes of real numbers on a negative basis
Part 2. Codes of complex numbers and vectors
Part 3. Codes of geometric shapes
Part 4. Codes of functions

4. DETAILED CONTENT

Foreword \ 0-21
References \ 0-24
Part 1. Codes of real numbers on a negative basis
Foreword \ 4
Chapter 1. Representation of numbers \ 1-6
1. Introduction \ 6
2. P-codes \ 7
3. M-codes \ 8
4. C-codes \ 9
Chapter 2. Algebraic addition of M-codes \ 1-11
1. Introduction \ 11
2. One-bit schemes of algebraic addition of M-codes \ 13
1. Inverter - Inv \ 13
2. Inverter - doubler - Inv2 \ 13
3. Inverse adder - InvAdd \ 14
4. Adder - Add \ 15

5. Subtractor - Sub \ 15
6. Algebraic adder - AddAlg \ 16
7. Subtractor with doubling - Sub2 \ 18
8. Inverse subtractor with additions - InvSubP \ 18
9. Subtractor with additions - SubP \ 21
10. Subtractor with reductions - SubM \ 22
11. Signifier - Seven, Sodd \ 23
3. Multi-bit schemes of algebraic addition of M-codes \ 26
1. Linear circuits \ 26
2. Sparse code subtractor with the addition of 2 - nSubP \ 28
3. Subtracting sparse codes with subtraction 2 - nSubM \ 28
4. Inverse subtracter of sparse codes with the addition of 2 - nInvSubP \ 29
5. Layered schemes \ 30
6. Three-tiered algebraic adder - nAddAlg3 \ 31
7. Three-tier adder - nAdd3 \ 31
8. Three-tier subtractor - nSub3 \ 31
9. Signifier - nSign \ 32
Chapter 3. Algebraic addition of C-codes \ 1-33
1. Multi-bit schemes of algebraic addition of C-codes \ 33
1. Algebraic adder mantises C-code - SumM2 \ 33
2. Complete algebraic adder mantises C-code - AddFull \ 35
2. Algebraic addition operations of C-codes \ 38
1. Unary operations with integer codes - Uni \ 38
2. Unary operations with non-integer codes - Unf \ 39
3. Binary operations with integer codes - Bini \ 40
4. Binary operations with memory overflow - BiniR \ 40
5. Binary operations, taking into account the previous overflow - BiniW \ 41
6. Binary operations, taking into account the previous overflow and remembering the overflow - BiniWR \ 41
7. Binary operations double-precision \ 41
8. Binary operations with non-integer codes - Binf \ 42
Chapter 4. Group Transfer \ 1-43
1. The theory of group transfer \ 43
0. Introduction \ 43
1. Definition of critical combinations \ 45
2. Synthesis of logical formulas for the transfer generation scheme \ 47
3. The equivalent circuit \ 49
2. Algebraic adders of M-codes with group transfer \ 52
1. Introduction \ 52
2. Inverse adder with odd group carry \ 53
3. Inverse adder with even group carry \ 55
4. Inverter with odd group transfer \ 56
5. Inverter with even group transfer \ 57
Chapter 5. Encoding and decoding \ 1-59
1. Coding and decoding devices for real numbers \ 59
1. Precoder of P-code in M-code - PreCoder \ 59
2. Coder of a positive P-code in the M-code - CoderPM \ 59
3. Precoder of P-code in the M-code with the addition - ExpCoder \ 61
4. Decoder M-code in the P-code - DecoderMP \ 62
5. The full decoder of the M-code in the P-code - mDecoderMP \ 64
6. Code Partitioner - Partitioning \ 64
2. The operations of encoding and decoding complex numbers \ 66
0. Introduction \ 66
1. Encoding a pair of integer P-codes into a whole C-code \ 66
2. Decoding an entire C-code into a pair of integer P-codes \ 67
3. Encoding a pair of P-codes into a pair of C-codes \ 69
4. Decoding a C-code into a pair of P-codes \ 70
5. Encoding a pair of P-codes in a C-code \ 71
7. Isolation of the integer and fractional parts of the C-code \ 71

8. Isolation of the integer part of the C-code \ 73
9. Isolation of the fractional part of the C-code \ 73
10. Convert exponent to integer C-code \ 73
11. Convert exponent to C-code \ 74
12. Converting the whole C-code to the exponent \ 74
13. Converting the C code to the whole C code \ 74
14. Converting the whole C code to C code \ 74

Chapter 6. Special logical operations \ 1-75
1. Normalization \ 76
2. Alignment of exponents \ 79
3. Shifts \ 81
    1. Mantissa shifter to the right - ShiftRigth \ 81
    2. Mantissa shift to the left - ShiftLeft \ 83
    3. Mantissa fast shifter to the right - ShFastR \ 83
    4. Mantissa fast shifter to the left - ShFastL \ 84
4. Comparators \ 87
    1. Block comparing C-codes modulo - CompModLfull \ 87
    2. Block comparing M-codes in length - CompLen \ 87
    3. Comparison operations \ 89
        1. Comparison in length - CompareLen \ 89
        2. Comparison modulo - CompareMod \ 89
        3. Comparison of the length of the codes of real and imaginary parts - CompareLenReIm \ 89
        4. Comparison modulo real and imaginary parts - CompareModReIm \ 89
        5. Comparison of the length of complex floating-point codes - CompareModFloat \ 90
        6. Comparison of exhibitors - CompareExp \ 90

Chapter 7. Multiplication \ 1-91
1. The method of multiplying M-codes \ 91
2. Matrix Multipliers \ 93
    1. Inverse matrix multiplier of M-codes - MultMatrShort \ 93
    2. Matrix quadrator of M-codes - QuadrMatr \ 96
    3. The matrix multiplier of M-codes - MultMatr \ 96
    4. The matrix multiplier of C-codes - FloatMultMatr \ 96
    5. Matrix quadrator of C-codes - CalcModul \ 97
3. Addition of the group of code \ 98
4. Formation of a group of terms in multiplication \ 103
5. Compound Multipliers \ 1098
    1. The multiplier of M-codes with the Wallis tree \ 108
    2. M-code multiplier with pre-compressor \ 109
    3. The multiplier of M-codes on the squares \ 110
    4. The multiplier of C-codes on the squares \ 110
6. The operations of the multiplication of M-codes and C-codes \ 112
    1. Multiplication of complex numbers \ 115
    2. The square of the complex number \ 115
    3. The square of the modulus of the complex number \ 115
    4. Centroaffine conversion \ 115
    7. Scalar multiplication of complex numbers \ 115
    11. Multiplication of integer complex numbers \ 116
    12. The square of the integer complex number \ 116
    13. The square of the modulus of the integer complex number \ 116
    14. Centroaffine transformation of an integer complex number \ 116
    15. Athenian conversion of an integer complex number \ 117
    16. Fast Fourier transform of integer complex numbers \ 117
    21. Multiplication of real numbers \ 117
    22. The square of the real number \ 118
    23. The determinant of the square matrix of real numbers \ 118
    24. Multiplication of a matrix by a vector for real numbers \ 118

31. Multiplication of integer real numbers \ 119
32. The square of the integer real number \ 119
35. Multiplication of four pairs of integer real numbers \ 119
Chapter 8. Division \ 1-20
1. Method of division \ 120
2. Decompositions and compositions \ 122
3. The algorithm of circulation and division \ 124

Part 2. Codes of complex numbers and vectors
Foreword \ 2-4
Chapter 1. Positional codes of complex numbers and vectors \ 2-8
1. About the method of positional coding \ 8
2. Two ways of synthesizing codes of complex numbers \ 10
3. The method of coding points of multidimensional space \ 12
4. Arithmetic coding systems \ 18
5. Codes of real numbers \ 21
6. Codes of complex numbers \ 23
7. Codes of multidimensional vectors \ 32
Chapter 2. Coding Accuracy \ 2-36
1. The range of representable numbers \ 36
2. The range of representable modules \ 44
3. Errors of final codes \ 51
4. Floating-Point Codes \ 54
Chapter 3. Bitwise arithmetic operations \ 2-59
1. Bitwise operations \ 59
2. The first algorithm for bitwise operations \ 61
3. The second algorithm of bitwise operations \ 68
4. Bitwise operations with a negative base \ 79
Chapter 4. Algorithms for encoding and decoding complex numbers \ 2-81
Chapter 5. Multiplication \ 2-85
1. Special algebra in the vector space \ 85
1.1. Algebra in three-dimensional vector space \ 85
1.2. Component Multiplication \ 86
1.3. Vector product \ 86
1.4. Scalar Product \ 86
1.5. Rotate the vector \ 87
1.6. Centroaffine Transformation \ 88
1.7. Multidimensional space \ 88
2. Multiplication of multidimensional vectors \ 89
2.1. Method of multiplication of complex numbers and multidimensional vectors \ 81
2.2. Vector multiplication by base (1.31) \ 82
2.3. Vector multiplication by base (1.30) \ 82
2.4. Sequential and matrix multiplication \ 83
2.5. Multiplication of integer codes of vectors by the base (1.31) \ 86
2.6. Multiplication of integer codes of vectors by the base (1.30) \ 86
2.7. Component-wise multiplication of multidimensional vectors \ 87
3. Scalar and vector multiplication \ 97
3.1. Scalar Product \ 97
3.2. Vector product \ 98
3.3. Carriers in scalar multiplication \ 99
3.4. Carriers in vector multiplication \ 101
Chapter 6. The method of "digit by digit" \ 2-103
1. Introduction \ 103
2. Decompositions \ 105
1. Introduction \ 105
2. The decomposition algorithm \ 106
3. Decomposition options \ 109
3. Compositions \ 114
4. Two-step operations \ 118
1. Introduction \ 118
2. Function calculation algorithm \ 118
3. About hardware implementation \ 119
5. Schemes and operations with binoms \ 122
1. Binoms multiplication schemes \ 122
2. Operations with binoms\ 125
6. Collector \ 126
1. Collector device \ 126
2. The block search triad \ 128
3. Logical block \ 129
4. Accelerated logic block \ 130
5. Three-digit logic circuit \ 131
6. One-bit logic circuit \ 132
7. Operations with the collector \ 133
1. Read H-discharge collector \ 133
2. Resetting the H-discharge of the collector \ 133
3. Adding "1" to the H-bit of the collector \ 133
4. The search of the first element of the collector \ 133

5. The search of the next collector element \ 133
6. Setting the maximum in the counter of collector \ 136.
Chapter 7. Division \ 2-134
1. Method of division \ 134
2. Division of complex numbers by base ρ_2 \ 137
3. The division of real numbers in the system with the base "-2" \ 144
4. Division of three-dimensional vectors \ 147
Chapter 8. Calculation of functions of a complex variable \ 2-155
1. Logarithm \ 155
1. Definition of the natural logarithm of a complex number (option 1) \ 155
2. Calculation of the logarithm of the module of a complex number \ 156
3. The definition of the natural logarithm of a complex number (option 2) \ 156
4. Definition of the natural logarithm of a positive real number \ 157
5. The definition of the natural logarithm of a negative real number \ 157
6. Definition of the natural logarithm of a real number \ 157
2. Potentiation \ 158
1. Potentiation of the complex number \ 158
2. Potentiation of the real number \ 160
3. Operations with logarithmic forms \ 162
1. Logarithmic representation of a complex number \ 162
2. Formation of the logarithmic form \ 162
3. Return from the logarithmic form \ 163
4. Algebraic addition of logarithmic forms \ 165
5. Multiplication of the logarithmic form by an integer \ 165
6. Overflow \ 165
4. Extraction of the square root \ 166
1. Extraction of the square root of the complex number \ 166
2. Extraction the square root of the conjugate number \ 166
3. Extraction the root from a positive real number \ 166
5. Polar coordinates \ 168
1. The calculation of the modulus of a complex number \ 168
2. The calculation of the argument of a complex number (option 1) \ 168
3. The calculation of the argument of a complex number (option 2) \ 168
4. Calculation of polar coordinates \ 169
5. Return from polar coordinates \ 170
6. Calculation of the sine and cosine of the real number \ 170
7. The definition of the semi-quadrant \ 172
8. Segmentation before root extraction \ 172
9. Segmentation after root extraction \ 173
10. Segmentation for logarithm \ 174
6. Operations with polar forms \ 176
1. The polar form of the complex number \ 176
2. Multiplication of exponential forms \ 176
3. The rotation of the exponential form \ 177
7. Complex functions \ 178
Chapter 9. The solution of the equations \ 181
1. The method of "digit by digit" and transcendental equations / 181
2. Determination of the roots of the power polynomial \ 185
3. Differential equations \ 189
4. The solution of quadratic equations \ 190
5. The solution of transcendental equations by the method of Mueller \ 192
Chapter 10. Modeling a device for extracting the square root of complex numbers \ 2-193
1. Introduction \ 193
2. Extraction of the square root \ 194
1. Compositions \ 194
2. Decompositions \ 196
3. Area of representation of numbers \ 200
4. Performance evaluation \ 205
5. Performance optimization \ 206
3. Sqrt - normalization \ 210
4. On convergence \ 215
5. Some comparisons \ 217
Chapter 11. Modeling of a device for potentiation and logarithm of complex numbers \ 2-220

1. Introduction \ 220
2. Composition \ 220
3. Decompositions \ 225
4. The region of representable numbers \ 229
5. Performance evaluation \ 236
6. Performance optimization \ 237
7. Additional functions of the device \ 240
1. The calculation of the orth of the complex number \ 240
2. Calculation of the logarithm of the module and the argument of the complex number \ 241
3. Calculation of the root of the square and logarithm of the complex number \ 241
Chapter 12. Modeling a device for computing one root of a power polynomial \ 2-242
1. Compositions \ 242
2. Decompositions for the solution of the normal square trinomial \ 245
3. Root - normalization when computing one root of a power polynomial \ 252
1. Introduction \ 252
2. Root-normalization by angle \ 253
3. Root-normalization modulo \ 253
4. Full root-normalization \ 254

Part 3. Codes of geometric shapes
Foreword \ 3-4
Chapter 1. Introduction \ 3-6
Chapter 2. Prototypes \ 3-10
1. Data Presentation \ 10
2. The simplest arithmetic device \ 11
3. The arithmetic device with rectangular codes \ 13
Chapter 3. The arithmetic of complex numbers and vectors \ 3-16
1. Introduction \ 16
2. Multiplication of multidimensional vectors \ 16
1. The method of multiplication of multidimensional and vectors \ 16
2. Multiplication by the basic function for vectors at the base (3.3.10) \ 17
3. Multiplication by the basic function for vectors at the base (3.3.7) \ 17
4. Multiplication of integer codes of vectors at the basis of (3.3.10) \ 18
5. Multiplication of integer codes of vectors on the basis of (3.3.7) \ 19
6. Component-wise multiplication of multidimensional vectors \ 20
7. Scalar and vector multiplication \ 21
3. Algorithms and devices for encoding and decoding multi-dimensional vectors \ 21
1. Coding a complex number in the system 1 \ 21
2. Decoding a complex number in the system 1 \ 22
3. Coding a complex number in the system 2 \ 22
4. Decoding a complex number in the system 2 \ 23
Chapter 4. Vector processor \ 3-24
1. Data representation and vector arithmetic device \ 24
2. Comparisons \ 28
Chapter 5. Theory of coding figures \ 3-31
1. Primary geometric codes \ 31
1. Data structure \ 31
2. Arithmetic operations with geometric codes on a real base \ 34
1. General \ 34
2. Writing the base code \ 35
3. Transpose \ 36
4. Addition of geometric and basic codes at the base (2) \ 36
5. Algebraic addition of geometric and basic codes at the base (2) \ 39
6. Algebraic addition of geometric and basic codes at the base (-2) \ 39
7. Multiplication of geometric and basic codes \ 42
8. Division of a geometric code into a base code \ 47
9. Rounding the geometric code \ 47
3. Geometric codes at a complex base \ 47
1. Algebraic addition of geometric and basic codes \ 48
2. Multiplication of geometric and basic codes \ 49
4. Coding and transformation of plane figures \ 53
1. Encoding method \ 53
2. Transfer \ 58
3. Centroaffine transformation \ 58
4. Affine transformation \ 58
5. Coding and transformation of spatial figures \ 59
2. Attribute geometric codes \ 61
1. Data structure \ 61
2. AGC on a real basis \ 65
1. Record of given number \ 65

2. Record and search of the given value \ 65
3. Reading the value of the path with the given number \ 66
4. Addition of AGC with the base code at the base (2) \ 66
5. The reverse addition of AGC with the base code at the base (-2) \ 68
6. Inverting AGC on the basis of (-2) \ 70
7. Algebraic addition of AGC \ 71
8. Search for the next open path, its number and its value \ 71
9. Multiplication of AGC by basis code \ 71
3. Attribute geometric codes on a complex base \ 72
1. Reverse addition of AGCC with the base code \ 73
2. Inverting \ 73
3. Centroaffine transformation \ 73
4. Attribute geometric codes of spatial figures \ 78
5. Abbreviated attribute geometric codes \ 81
Chapter 6. Geometric processor \ 3-83
0. Data Presentation \ 83
1. Full specialized random access memory \ 86
2. Fragmented specialized random access memory \ 87
3. The maximum arithmetic device of geometric figures \ 90
4. The fragmented arithmetic device of geometric figures \ 91
5. The processor with the maximum arithmetic device \ 93
6. Processor with fragmentary arithmetic device \ 95
7. Basic procedures \ 98
1. Affine transformation \ 98
2. Rounding \ 99
3. Rough rounding \ 100
4. Correction of attributes \ 101
5. Calculation of attributes \ 101
6. Coding a figure \ 102
7. Decoding a figure \ 102
8. Operation blocks \ 102
1. The unit for recording the number with given code \ 103
2. The unit for writing values with given code \ 103
3. The unit for reading the value of the path with the given number \ 104
4. Reverse adder \ 105
5. Search block of the first open path \ 106
6. Block for reading the number and value of the path with a given terminal vertex \ 107
7. Search block for the next terminal vertex \ 108
Chapter 7. Comparative analysis \ 3-109
Designations \ 3-115
List of examples \ 3-118
List of tables \ 3-119
List of drawings \ 3-121

Part 4. Codes of functions \ 4-1
Foreword \ 4-4
Chapter 1. Positional codes of functions \ 4-6
1. Triangular codes \ 6
2. Algebraic addition of codes of real numbers \ 10
3. Algebraic addition of triangular codes \ 12
4. Division of triangular codes on parameter \ 14
5. Multiplication of triangular codes \ 16
6. Encoding and decoding of triangular codes \ 17
7. Differentiation of triangular codes \ 19
8. Stepped codes \ 21
Chapter 2. Coding of trigonometric series \ 4-23
1. Triangular codes of functions on base $Sin^2(x)$ \ 23
2. Trigonometric triangular codes - TTC \ 36
3. Operations with TTC \ 40
1. Short operations \ 40
2. Multiplication \ 40
3. Differentiation \ 41
4. Integration \ 43
5. Inverting the argument \ 44
6. The shift of the ordinate axis \ 45
4. Encoding and decoding of TTC\ 46
5. The encoding error TTC \ 48
6. Shortening TTC \ 52
7. Hyperbolic triangular codes \ 55
Chapter 3. Coding of functions of many arguments \ 4-56
1. Pyramidal codes \ 56
2. Hyperpyramidal codes \ 59
Chapter 4. Quaternary trigonometric triangular codes \ 4-62
1. Arithmetic operations with quaternary triangular codes \ 64
1. Algebraic addition \ 64

2. Division on parameter \ 66
3. Multiplication \ 70
2. Encoding and decoding of quaternary triangular codes \ 71
1. Encoding and decoding of quaternary triangular codes \ 71
2. Encoding and decoding of quaternary trigonometric triangular codes \ 76
3. Mathematical operations with quaternary trigonometric triangular codes \ 85
4. Shortening of quaternary trigonometric triangular codes \ 86
5. The coding error of quaternary trigonometric triangular codes \ 91
Chapter 5. Arithmetic device for operations with functions \ 4-95
1. Single-bit circuits \ 95
1. One-digit adder \ 95
2. One-digit subtractor \ 96
3. One-bit inverter \ 97
4. One-digit quadrupler \ 98
5. One-digit divider on 4 \ 98
2. Multi-bit circuits \ 99
1. Adder of columns\ 99
2. Line adder \ 100
3. Divider of columns \ 101
4. Line divider \ 102
5. Parallel adder \ 102
6. Parallel divider \ 103
3. Option arithmetic device \ 104
1. The structure of the arithmetic device \ 104
2. Operations with triangular codes \ 106
1. Algebraic addition of mixed codes \ 106
2. Quadruple mixed code \ 106
3. Division of the triangular code by 4 \ 106
5. Rounding off a triangular code \ 106
6. Convert rectangular code to triangular code \ 107
7. Converting a triangular code to a rectangular code \ 108
8. Multiplication of triangular codes \ 108
3. Operations with trigonometric triangular codes \ 109
1. Algebraic addition of TTK \ 109
2. The multiplication of TTK \ 109
3. Differentiation of TTK \ 109
4. Integration of TTK \ 110.
5. Coding of trigonometric series \ 110
6. Decoding TTK \ 111
7. Shortening TTK \ 111
4. Comparative analysis \ 111
1. The relationship between the capacity of the TTK, the rank of the series and the capacity of the coefficients of the series \ 112
2. Capacity \ 113
3. The volume of the arithmetic device \ 113
4. The duration of elementary operations \ 114
5. The relationship between elementary operations and operations with functions \ 115
6. Conclusions \ 116
Designations \ 118

5. REFERENCES

[1]. Solomon I. Khmelnik. Computer Arithmetic of Numbers, Vectors, Figures and Functions. Algorithms and Hardware, http://doi.org/10.5281/zenodo.1309433 (by Ruusian)

