THE PAPERS OF INDEPENDENT AUTHORS

Biology
Computer Engineering
Geology
Physics and Astronomy
ADS

Eckardt Horst
Etkin Valery A.
Felker Laurence G.
Gershman Yakov H.
Khmelnik Solomon I.
Korchazhkin Sergey V.
Vilshanski Alexander N.
Zamaliev Pavel S.

The Papers of Independent Authors, volume 28

Series: **COMPUTER ENGINEERING**

Solomon I. Khmelnik

Computer for operations with functions

Annotation

A computer for operations with (mathematical) functions (unlike the usual computer) operated with function at the hardware level (i.e. without programming these operations). The article briefly describes the functions of one and many arguments coding theory - the codes structure, algorithms of coding and decoding, arithmetical operations. Processor for operations with functions considered - data representation, operating blocks, hardware realization of coding, decoding and arithmetic operations algorithms.

Contents

1. History

A computing machine for operations with functions was presented and developed by Mikhail Kartsev in 1967 [1]. Among the operations of this computing machine were the functions addition, subtraction and multiplication, functions comparison, the same operations between a function and a number, finding the function maximum,

computing indefinite integral, computing definite integral of derivative of two functions, derivative of two functions, shift of a function along the X-axis etc. By its architecture this computing machine was (using the modern terminology) a vector processor or *array processor*, a central processing unit (CPU) that implements an instruction set containing instructions that operate on one-dimensional arrays of data called *vectors*. In it there has been used the fact that many of these operations may be interpreted as the known operation on vectors: addition and subtraction of functions - as addition and subtraction of vectors, computing a definite integral of two functions derivative— as computing the vector product of two vectors, function shift along the X-axis – as vector rotation about axes, etc. [1] In 1966 Khmelnik had proposed a functions coding method [2], i.e. the functions representation by a "uniform" (for a function as a whole) positional code. And so the mentioned operations with functions are performed as unique computer operations with such codes on a "single" arithmetic unit [3].

2. Positional codes of one-variable functions [2, 3]
2.1. The main idea

The positional code of an integer number A is a numeral notation of digits α in a certain <u>positional number system</u> of the form

$$A = \alpha_0 \alpha_1 \ldots \alpha_k \ldots \alpha_n.$$

Such code may be called "linear". Unlike it a positional code of one-variable x function $F(x)$ has the form:

$$F(x) = \begin{pmatrix} & & & \cdots & \\ & & \alpha_{22} & \cdots & \alpha_{2k} & \cdots \\ & \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1k} & \cdots \\ \alpha_{00} & \alpha_{01} & \alpha_{02} & \cdots & \alpha_{0k} & \cdots \end{pmatrix}$$

and so it is *flat* and "triangular", as the digits in it comprise a triangle. The value of the positional number A above is that of the sum

$$A = \sum_{k=0}^{n} \alpha_k \rho^k,$$

where ρ is the radix of the said number system. The positional code of a one-variable function correspond to a 'double' code of the form

$$F(x) = \sum_{k=0}^{n} \sum_{m=0}^{k} \alpha_{mk} R^k y^{k-m} (1-y)^m,$$

where R is an integer positive number, quantity of values that taken α, and y is a certain function of argument x.

Addition of positional codes of numbers is associated with the <u>carry</u> transfer to a higher digit according to the scheme

$$\alpha_k \longrightarrow \alpha_{k+1}.$$

Addition of positional codes of one-variable functions is also associated with the carry transfer to higher digits according to the scheme:

$$\begin{pmatrix} & & \alpha_{k+1,m+1} \\ & \nearrow & \\ \alpha_{k,m} & \longrightarrow & \alpha_{k+1,m} \end{pmatrix}.$$

Here the same transfer is carried simultaneously to *two* higher digits.

2.2. R-nary triangular code

A triangular code is called R-*nary* (and is denoted as TK_R), if the numbers α_{mk} take their values from the set

$$D_R = \{-r_1, -r_1 + 1, \ldots, -1, 0, 1, \ldots, r_2 - 1, r_2\},$$

where $R = r_1 + r_2 + 1$, $r_1 > 0$, $r_2 > 0$. For example, a triangular code is a ternary code TK_3, if $\alpha_{mk} \in \{-1,0,1\}$, and quaternary TK_4, if $\alpha_{mk} \in \{-2,-1,0,1\}$. For R-nary triangular codes the following equalities are valid:

$$\begin{bmatrix} & 0 \\ Ra & 0 \end{bmatrix} = \begin{bmatrix} & a \\ 0 & a \end{bmatrix}, \quad \begin{bmatrix} & a \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} & 0 \\ Ra & -a \end{bmatrix}, \quad \begin{bmatrix} & 0 \\ 0 & a \end{bmatrix} = \begin{bmatrix} & -a \\ Ra & 0 \end{bmatrix}.$$

where a is an arbitrary number. There exists TK_R of an arbitrary integer real number. In particular, $TK_R(\alpha) = \alpha$. Also there exists TK_R of any function of the form y^k. For instance, $TK_R(y^2) = (0\ 0\ 1)$.

2.2.1. Single-digit addition
in R-nary triangular codes consists in the following:
- in the given (m,k)-digit there is determined the sum S_{mk} of the digits that are being added α_{mk}, β_{mk} and two carries $P_{m,k-1}, P_{m-1,k-1}$, transferred into this digit from the left, i.e.

$$S_{m,k} = \alpha_{m,k} + \beta_{m,k} + p_{m,k-1} + p_{m-1,k-1},$$

- this sum is presented in the form $S_{mk} = \sigma_{mk} + Rp_{mk}$, where $\sigma_{mk} \in D_R$,

- σ_{mk} is written in the (m,k)-digit of summary code, and the carry p_{mk} from the given digit is carried into (m,k+1)-digit and (m+1,k+1)-digit.

This procedure is described (as also for one-digit addition of the numbers) by a table of one-digit addition, where all the values of the terms $\sigma_{mk} \in D_R$ and $\beta_{mk} \in D_R$ must be present and all the values of carries appearing at decomposition of the sum $S_{mk} = \sigma_{mk} + Rp_{mk}$. Such a table may be synthesized for $R>2$. Below we have written the table of one-digit addition for $R=3$:

Smk	TK(Smk)	σ_{mk}	p_{mk}	
.	.	0	.	.
0	0	0	0	0
.	.	0	.	.
1	1	0	1	0
.	.	0	.	.
(-1)	(-1)	0	(-1)	0
.	.	1	.	.
2	(-1)	1	(-1)	1
.	.	1	.	.
3	0	1	0	1
.	.	1	.	.
4	1	1	1	1
.	.	(-1)	.	.
(-2)	1	(-1)	1	(-1)
.	.	(-1)	.	.
(-3)	0	(-1)	0	(-1)
.	.	(-1)	.	.
(-4)	(-1)	(-1)	(-1)	(-1)

2.2.1. One-digit subtraction
in R-nary triangular codes differs from the one-digit addition only by the fact that in the given (m,k)-digit the value $S_{m,k}$, is determined by the formula

$$S_{m,k} = \alpha_{m,k} - \beta_{m,k} + p_{m,k-1} + p_{m-1,k-1},$$

2.2.3. One-digit division by the parameter R
in R-nary triangular codes is based on using the correlation:

$$\begin{pmatrix} & a \\ & \nearrow & \\ 0 \longrightarrow & 0 \end{pmatrix} = \begin{pmatrix} & 0 \\ & \nearrow & \\ aR \longrightarrow & -a \end{pmatrix},$$

from this it follows that the division of each digit causes carries into two lowest digits. Hence, the digits result in this operation is a sum of the quotient from the division of this digit by R and two carries from two highest digits. Thus, when divided by parameter R

- in the given (m,k)-digit the following sum is determined

$$S_{m,k} = \alpha_{m,k}/R - p_{m+1,k}/R + p_{m+1,k+1},$$

- this sum is presented as $S_{m,k} = \sigma_{m,k} + p_{m,k}/R$, where $\sigma_{mk} \in D_R$,

- σ_{mk} is written into (m,k)-digit of the resulting code, and carry $p_{m,k}$ from the given digit is transferred into the (m-1,k-1)-digit and (m-1,k)-digit.

This procedure is described by the table of one-digit division by parameter R, where all the values of terms and all values of carries, appearing at the decomposition of the sum $S_{m,k} = \sigma_{m,k} + p_{m,k}/R$, must be present. Such table may be synthesized for $R>2$. Below the table will be given for the one-digit division by the parameter R for $R=3$:

Smk	TK(Smk)	σ_{mk}	p_{mk}
.	.	0	.	.
0	0	0	0	0
.	.	1	.	.
1	0	0	1	0
.	.	(-1)	.	.
(-1)	0	0	(-1)	0
.	.	0	.	.
1/3	1	(-1/3)	0	1
.	.	1	.	.
2/3	(-1)	1/3	1	(-1)
.	.	1	.	.
4/3	1	(-1/3)	1	1
.	.	2	.	.
5/3	(-1)	1/3	2	(-1)
.	.	0	.	.
(-1/3)	(-1)	1/3	0	(-1)
.	.	(-1)	.	.
(-2/3)	1	(-1/3)	(-1)	1
.	.	(-1)	.	.
(-4/3)	(-1)	1/3	(-1)	(-1)
.	.	(-2)	.	.
(-5/3)	1	(-1/3)	(-2)	1

2.2.4. Addition and subtraction

of R-nary triangular codes consists (as in positional codes of numbers) in subsequently performed one-digit operations. Mind that the one-digit operations in all digits of each column are performed simultaneously.

2.2.5. Multiplication

of R-nary triangular codes. Multiplication of a code TK'_R by (m,k) - digit of another code TK''_R consists in (m,k)-shift of the code TK'_R, i.e. its shift k columns left and m rows up. Multiplication of codes TK'_R and TK''_R consists in subsequent (m,k)-shifts of the code TK'_R and addition of the shifted code TK'_R with the part-product (as in the positional codes of numbers).

2.2.6. Derivation

of R-nary triangular codes. The derivative of function $F(x)$, defined above, is

$$\frac{\partial F(x)}{\partial x} = \frac{\partial y}{\partial x}\frac{\partial F(x)}{\partial y}.$$

So the derivation of triangular codes of a function $F(x)$ consists in determining the triangular code of the partial derivative $\partial F(x)/\partial y$ and its multiplication by the known triangular code of the derivative $\partial F(x)/\partial y$ $\partial y/\partial x$. The determination of the triangular code of the partial derivative $\partial F(x)/\partial y$ is based on the correlation

$$\frac{\partial}{\partial x}\begin{pmatrix} & & 0 \\ & 0 & \alpha_{mk} \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} & & (k-m)\alpha_{mk} \\ & 0 & (k-2m)\alpha_{mk} \\ 0 & 0 & (-m)\alpha_{mk} \end{pmatrix}.$$

The derivation method consists of organizing carries from (m,k)-digit into (m+1, k)-digit and into (m-1, k)-digit, and their summing in the given digit is performed in the same way as in one-digit addition.

2.2.7. Coding and decoding

of R-nary triangular codes. A function represented by series of the form

$$F(x) = \sum_{k=0}^{n} A_k y^k,$$

with integer coefficients A_k, may be represented by R-nary triangular codes, for these coefficients and functions y^k have R-nary triangular codes (which was mentioned in the beginning of the section). On the

31

other hand, R-nary triangular code may be represented by the said series, as any term $\alpha_{mk} R^k y^k (1-y)^m$ in the positional expansion of the function (corresponding to this code) may be represented by a similar series.

2.2.8. Truncation

of R-nary triangular codes. This is the name of an operation of reducing the number of "non"-zero columns. The necessity of truncation appears at the emergence of carries beyond the digit net. The truncation consists in division by parameter R. All coefficients of the series represented by the code are reduced R times, and the fractional parts of these coefficients are discarded. The first term of the series is also discarded. Such reduction is acceptable if it is known that the series of functions converge. Truncation consists in subsequently performed one-digit operations of division by parameter R. The one-digit operations in all the digits of a row are performed simultaneously, and the carries from lower row are discarded.

2.3. Scale factor

R-nary triangular code is accompanied by a scale factor M, similar to exponent for floating-point number. Factor M permits to display all coefficients of the coded series as integer numbers. Factor M is multiplied by R at the code truncation. For addition factors M are aligned, to do so one of added codes must be truncated. For multiplication the factors M are also multiplied.

3. Positional code for functions of many variables [4]

Positional code for function of two variables is depicted on Figure 1. It corresponds to a "triple" sum of the form:

$$F(x,v) = \sum_{k=0}^{n} \sum_{m1=0}^{k} \sum_{m2=0}^{k} \alpha_{m1,m2,k} R^k y^{k-m1} (1-y)^{m1} z^{k-m2} (1-z)^{m2},$$

where R is an integer positive number, number of values of the figure $\alpha_{m_1,m_2,k}$, and $y(x)$, $z(v)$ - certain functions of arguments x, v correspondingly. On Figure 1 the nodes correspond to digits $\alpha_{m_1,m_2,k}$, and in the circles the values of indexes $m1$, $m2$, k of the

corresponding digit are shown. The positional code of the function of two variables is called "pyramidal". Positional code is called R-nary (and is denoted as PK_R), if the numbers $\alpha_{m_1,m_2,k}$ assume the values from the set D_R. At the addition of the codes PK_R the carry extends to four digits and hence $R \geq 7$.
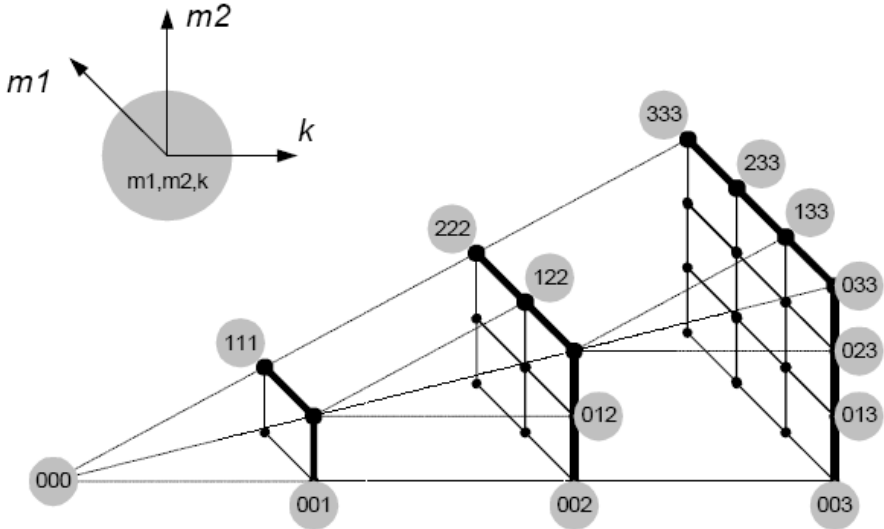

Fig. 1.

A positional code for the function from several variables corresponds to a sum of the form

$$F(x_1,\ldots,x_i,\ldots,x_a) = \sum_{k=0}^{n}\sum_{m_1=0}^{k}\ldots\sum_{m_a=0}^{k}\left(\alpha_{m_1,\ldots,m_a,k}R^k\prod_{i=1}^{a}(y_i^{k-m_i}(1-y_i)^{m_i})\right),$$

where R is an integer positive number, number of values of the digit $\alpha_{m_1,\ldots,m_a,k}$, and $y_i(x_i)$ certain functions of arguments x_i. A positional code of a function of several variables is called "hyperpyramidal". Of Figure 2 is depicted for example a positional hyperpyramidal code of a function of three variables. On it the nodes correspond to the digits $\alpha_{m_1,m_2,m_3,k}$, and the circles contain the values of indexes m_1, m_2, m_3, k of the corresponding digit. A positional hyperpyramidal code is called R-nary (and is denoted as GPK_R), if the numbers $\alpha_{m_1,\ldots,m_a,k}$ assume the values from the set D_R. At the

codes addition GPK_R the carry extends on a-dimensional cube, containing 2^a digits, and hence $R \geq \left(2^{a-1} - 1\right)$.
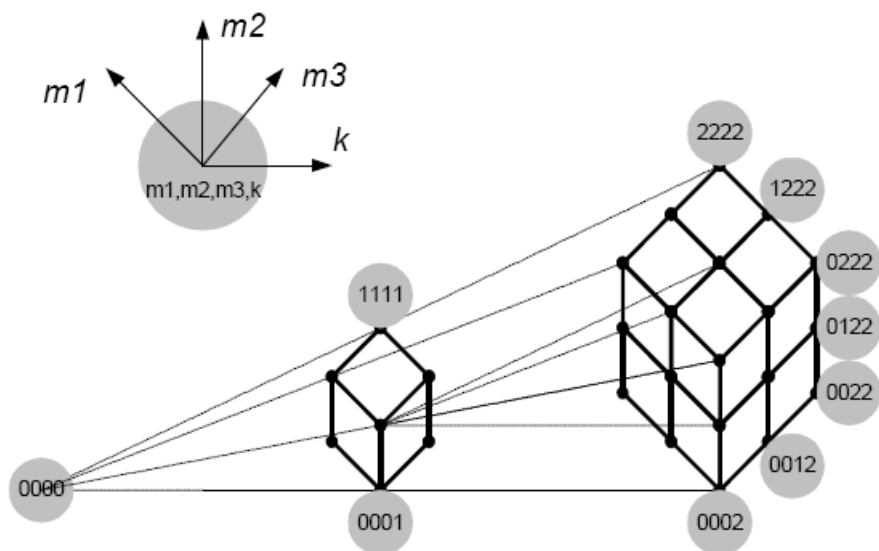

Fig. 2.

References

1. B.N. Malinovsky. The history of computer technology in their faces (in Russian). Kiew: Firm "KIT", 1995, ISBN 5-7707-6131-8, http://lib.ru/MEMUARY/MALINOWSKIJ/0.txt (in Russian), http://www.sigcis.org/files/SIGCISMC2010_001.pdf (in English)
2. S.I. Khmelnik. Coding of functions. Cybernetics, USSR Academy of Sciences, 4, 1966, http://lib.izdatelstwo.com/Papers2/s7.pdf (in Russian)
3. S.I. Khmelnik. Computer Arithmetic of Functions. Algorithms and Hardware Design. 2004, Israel: "Mathematics in Computers". ISBN 978-0-557-07520-1 (in Russian)
4. S.I. Khmelnik. Several types of positional functions codes. Cybernetics, USSR Academy of Sciences, 5, 1970, http://lib.izdatelstwo.com/Papers2/s17.pdf (in Russian)
5. http://en.wikipedia.org/wiki/Computer_for_operations_with_functions