УДК 681. 3

O.A. Finko

Algorithms and Devices for *N*-ary Finite Ring Computations

(Krasnodar high military school (military institute), Russia)

Methods of computer algebra connected with execution N-ary (N > 2) operations in a finite ring for specialized computing devices which operate in the modular arithmetics is developed. Is displayed that traditional methods of execution of N-ary operations which grounded on a "horizontal" method are oriented on a small amount of operands. Higher outcomes can be obtained with help parallel and sequential are N-ary of arithmetic devices which have grounded on a "vertical" method.

1. Introduction

The specialized computing devices (SCD) are applied to implementation of algorithms of digital signals processing (number-theoretic conversions, of cyclic convolution) [1—4], cryptography problems [5], control and simulations problems [6, 7], neural-like networks problems [8]. The important stage of development of SCD is multisequencing computations with the help of the modular arithmetics (MA) [10—12].

Today main attention of the experts is directed on increase of efficiency of execution of not modular operations [13, 14] and development of methods of a check and reconfiguration of SCD by redundant MA [15]. The following stage of development of SCD, which operate in the MA, can be integration of operations, which are realized in a parallel way. The integration of operations is supposed to be realized by transition from binary to *N*-ary modular operations. However now methods of implementation of *N*-ary operations on the given the modulo are advanced insufficiently. It is an obstacle in paths of implementation of the second stage of development of SCD. Therefore purpose of paper is to develop the effectiveness of algorithms and devices which are intended for implementation of *N*-ary operations of MA.

On the basis of Chinese remainder theorem [10—12] any integer $X \geq 0$ can be represented by a sequence of bit digits

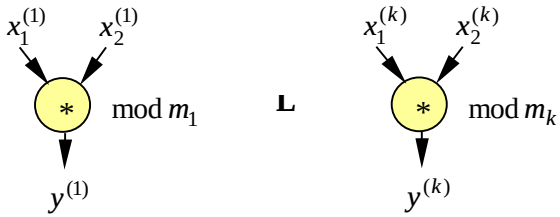
$$\{X\} = \left(x^{(1)}, x^{(2)}, \mathbf{K}, x^{(k)}\right),$$

where $x^{(i)} = |X|_{m_i}$, $i = 1, 2, \mathbf{K}, k$; $m_1, m_2, \mathbf{K}, m_k$ $\left(m_1 < m_2 < \mathbf{K} < m_k\right)$ — basis.

The representation of the MA is unique, if $0 \leq X < M = m_1 m_2 \, \mathbf{K} \, m_k$ and $\gcd\left(m_i, m_j\right) = 1$ for $\forall i \neq j, \quad i, j = 1, 2, \mathbf{K}, k$.

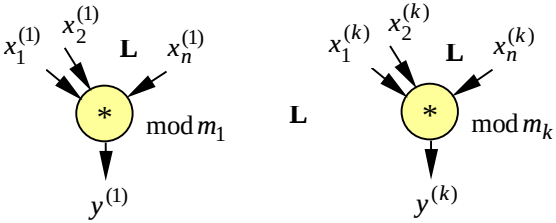
The operations of arithmetics MA are fulfilled by a parallel way and separately for each unit MA and therefore faster, than in traditional arithmetics, in which there are interbit links.

The existing algorithms MA use binary operations $x^{(i)} = \left|x_1^{(i)} * x_2^{(i)}\right|_{m_i}$ $\left(i = 1, 2, \mathbf{K}, k\right)$:



Here symbol * means one of modular operations (addition or multiplying modulo m_i).

The N-ary operation modulo m_i we shall name the operation $y^{(i)} = \left| \underset{j=1}{\overset{n}{*}} x_j^{(i)} \right|_{m_i}$:



The residuals $x^{(1)}, x^{(2)}, \mathbf{K}, x^{(k)}$ are represented with the help of a binary number system or unitary code. Let's consider a case of usage of a binary number system:

$$x_j^{(i)} = a_{d-1,\,j}^{(i)} 2^{k-1} + \mathbf{K} + a_{1,\,j}^{(i)} 2 + a_{0,\,j}^{(i)},$$

where $a_{r,\,j}^{(i)} \in \{0, 1\}, \;\; i = 1, 2, \mathbf{K}, k; \;\; r = 0, 1, \mathbf{K}, d-1; \qquad j = 1, 2, \mathbf{K}, n.$

Here the value of the module m_i and bit grid d are connected as $d = \lceil \log_2 (m_i - 1) \rceil$, where $\lceil A \rceil$ means the least integer $> A$.

2. *N*-ary operator of addition modulo m_i

2.1. Traditional architectures of *N*-ary summators modulo m_i

2.1.1. The "horizontal" *N*-ary summator modulo m_i of a parallel type. The structure of the traditional N-ary summator modulo m_i of a parallel type has a tree structure with $\lceil \log_2 (n-1) \rceil$ levels (figure 1) and $n-1$ by binary summators.

Hold time of this summator $T_\Sigma = \lceil \log_2(n-1) \rceil t_\Sigma$, where t_Σ time of operation of one binary summator modulo m_i.

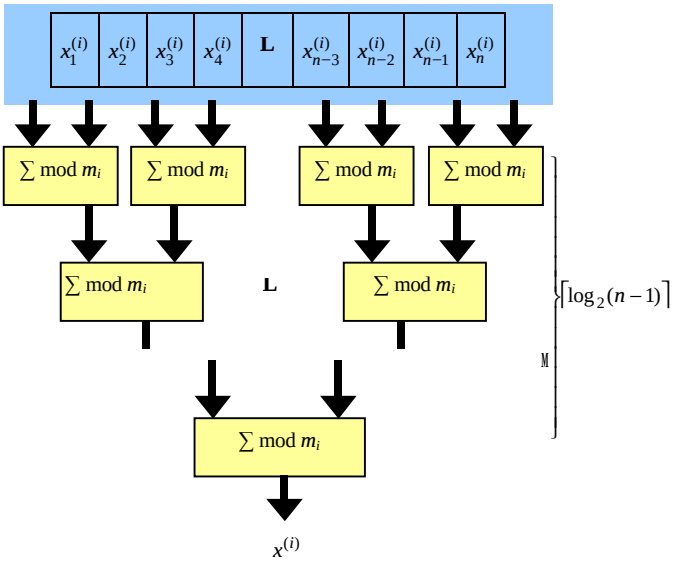


Figure 1. The "horizontal" *N*-ary summator modulo m_i of a parallel type

2.1.2. The "horizontal" *N*-ary summator modulo m_i of a sequential type. The traditional *N*-ary summator of a sequential type consists of the parallel-sequential register of shift and accumulating binary summator modulo m_i (figure 2). Amount of sync signals necessary for execution for the operation

$$y^{(i)} = \left| \sum_{j=1}^{n} x_j^{(i)} \right|_{m_i}$$

corresponds to total of addends and equally n.

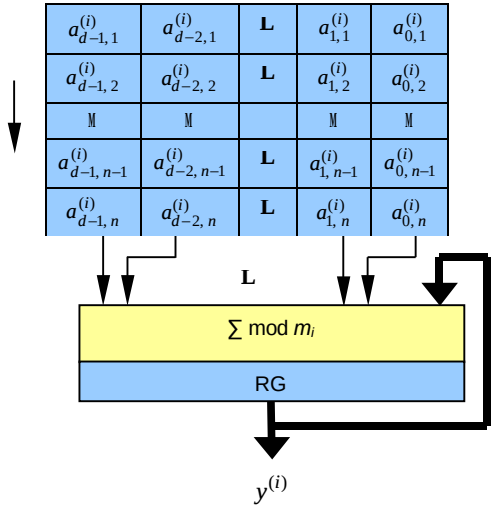


Figure 2. "Horizontal" *N*-ary summator modulo m_i of a sequential type

2.2. Algorithms and devices of *N*-ary addition modulo m_i, grounded on a "vertical" method

2.2.1. The algorithm of "vertical" *N*-ary addition modulo m_i. Let's consider algorithm *N*-ary addition module m_i, which works not with numbers $x_j^{(i)}$ $(j = 1, 2, \mathbf{K}, n)$, but with digits $a_{r,1}^{(i)}, a_{r,2}^{(i)}, \mathbf{K}, a_{r,n}^{(i)}$ $(r = d-1, \mathbf{K}, 1, 0)$.

Algorithm 2.1.

Step 1. Are realized d of count operations of units modulo m_i:

$$x_{d-1}^{(i)} = \left| \sum_{j=1}^{n} a_{d-1,j}^{(i)} \right|_{m_i}, \quad \mathbf{K}, \quad x_1^{(i)} = \left| \sum_{j=1}^{n} a_{1,j}^{(i)} \right|_{m_i}, \quad x_0^{(i)} = \left| \sum_{j=1}^{n} a_{0,j}^{(i)} \right|_{m_i}.$$

Step. 2. Realizing modular products of values $x_r^{(i)}$ $(r = 0, 1, \mathbf{K}, d-1)$ on 2^r $(r = 0, 1, \mathbf{K}, d-1)$:

$$\tilde{x}_{d-1}^{(i)} = \left| x_{d-1}^{(i)} 2^{d-1} \right|_{m_i}, \ \ldots, \ \tilde{x}_1^{(i)} = \left| x_1^{(i)} 2 \right|_{m_i}, \ \tilde{x}_0^{(i)} = x_0^{(i)}.$$

Step 3. The outcome of *N*-ary addition is:

$$y^{(i)} = \left| \sum_{j=0}^{d-1} \tilde{x}_r^{(i)} \right|_{m_i}.$$

2.2.2. The "vertical" *N*-ary summator modulo m_i of a parallel type. The structure of algorithm 2.1 can be presented as follows:

The device for parallel implementation of algorithm 2.1 (figure 3)

$$\begin{array}{lllll}
x_1^{(i)} = (a_{d-1,1}^{(i)}, & a_{d-2,1}^{(i)}, & \mathbf{L}, a_{1,1}^{(i)}, & a_{0,1}^{(i)}) \\
x_2^{(i)} = (a_{d-1,2}^{(i)}, & a_{d-2,2}^{(i)}, & \mathbf{L}, a_{1,2}^{(i)}, & a_{0,2}^{(i)}) \\
\mathbb{M} \quad\quad \mathbb{M} & \mathbb{M} & \quad\quad \mathbb{M} & \mathbb{M} \\
x_{n-1}^{(i)} = (a_{d-1,n-1}^{(i)}, & a_{d-2,n-1}^{(i)}, & \mathbf{L}, a_{1,n-1}^{(i)}, & a_{0,n-1}^{(i)}) \\
x_n^{(i)} = (a_{d-1,n}^{(i)}, & a_{d-2,n}^{(i)}, & \mathbf{L}, a_{1,n}^{(i)}, & a_{0,n}^{(i)}) \\
\hline
x_{d-1}^{(i)} & x_{d-2}^{(i)} & \mathbf{L} \ x_1^{(i)} & x_0^{(i)} \\
\downarrow & \downarrow & \quad\quad \downarrow & \downarrow \\
\tilde{x}_{d-1}^{(i)} & \tilde{x}_{d-2}^{(i)} & \mathbf{L} \ \tilde{x}_1^{(i)} & \tilde{x}_0^{(i)}
\end{array}$$

$$\searrow \quad \swarrow \qquad \searrow \quad \swarrow$$

$$\left| \tilde{x}_{d-1}^{(i)} + \tilde{x}_{d-2}^{(i)} \right|_{m_i} \qquad \left| \tilde{x}_1^{(i)} + \tilde{x}_0^{(i)} \right|_{m_i}$$

$$\searrow \qquad \swarrow$$

$$y^{(i)} = \left| \sum_{r=0}^{d-1} \tilde{x}_r^{(i)} \right|_{m_i}$$

contains the register of storage of binary bit digits, d of parallel counters of units (devices of convolution) ST $\bmod m_i$ and $d-1$ of binary summators $\Sigma \bmod m_i$. If on outputs of parallel units counters the unitary code, the additional encoders CD-1 … CD-*d* of the unitary code in the code of a binary number system are installed. For the usually used

module $2 \le m_i < 128$ a tree of binary summators — 1 … 3 levels. The main device of the considered summator — counter of units ST $\bmod m_i$. The principles of construction of such devices are well known [16—18], however, probably best outcomes were obtained in [19, 20].

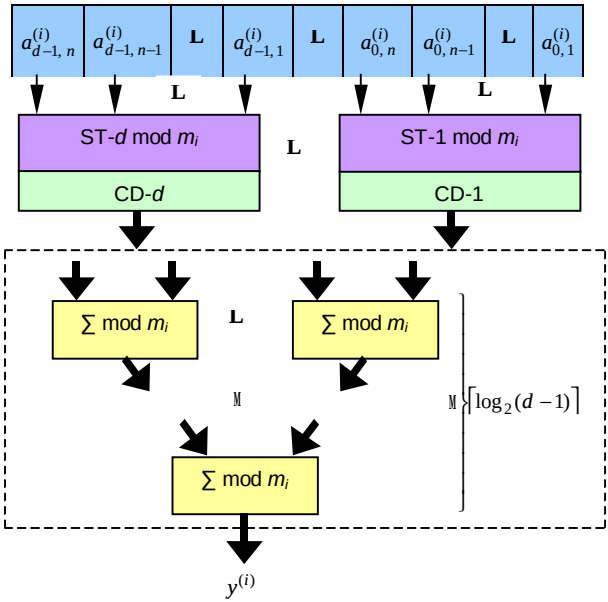
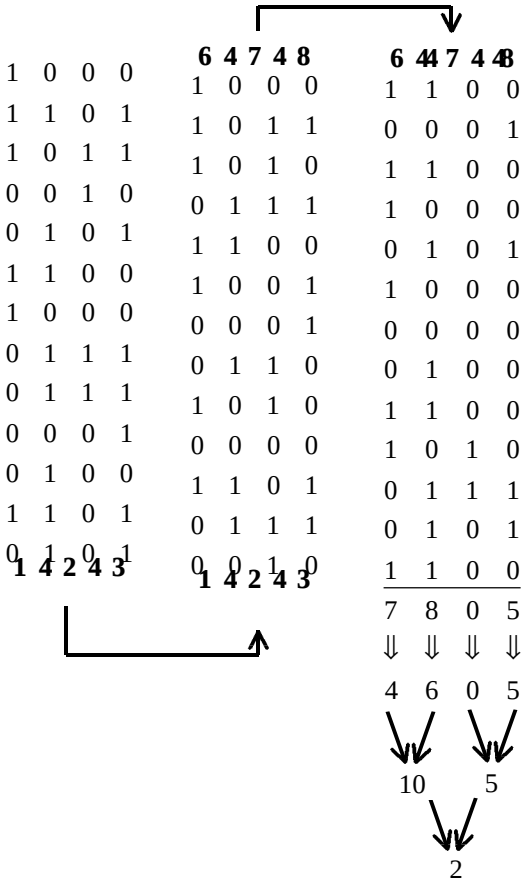


Figure 3. The "vertical" summator modulo m_i

Numerical example of addition of 39 operands modulo 13:



2.2.3. The "vertical" *N*-ary summator modulo m_i of a sequential type. We use numbers $x_{d-1}^{(i)}, \mathbf{K}, x_1^{(i)}, x_0^{(i)}$, from algorithm 2.1 for construction of the formula (principle of Horner):

$$y^{(i)} = \left| x_{d-1}^{(i)} 2^{d-1} + x_{d-2}^{(i)} 2^{d-2} + \mathbf{K} + x_1^{(i)} 2 + x_0^{(i)} \right|_{m_i} =$$

$$= \left| \left| \left(\mathbf{K} \left| \left(\left| \left(\left| x_{d-1}^{(i)} 2 \right|_{m_i} + x_{d-2}^{(i)} \right) 2 \right|_{m_i} + x_{d-3}^{(i)} \right) 2 \right|_{m_i} + \mathbf{K} + x_1^{(i)} \right) 2 \right|_{m_i} + x_0^{(i)} \right|_{m_i} .$$

.

The summator realizing this formula contains the parallel-sequential register of shift the counter of units ST mod m_i in the unitary code (encoder CD is not used) multiplying tube on 2 modulo m_i and binary adderaccumulator modulo m_i (figure 4).

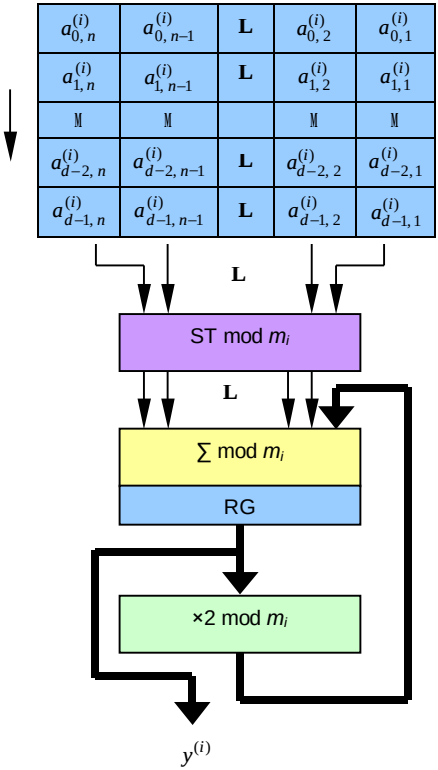


Figure 4. The "vertical" *N*-ary summator modulo m_i of a sequential type

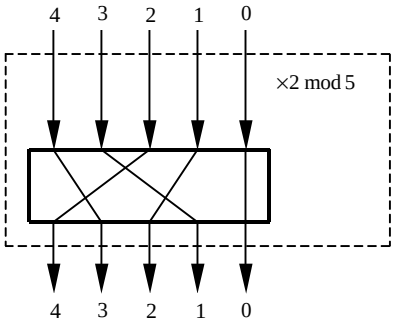


Figure 5. Structure of the multiplying tube on 2 modulo m_i

The principle of operation of *N*-ary summator is completely defined by the last obtained formula. The last clock tick of operation of the summator is used for addition $x_0^{(i)}$.

The unitary code is convenient for execution of the operation of multiplying on a constant in a finite ring. Thus the operation of multiplying is defined by the scheme of swaps of bits. The example of construction of the multiplying tube modulo 5 operation using this principle represented on a figure 5.

3. "Vertical" *N*-ary the converter of the code of a binary number system in the code modulo m_i

3.1. Algorithm of "vertical" *N*-ary conversion of a binary number system in the code modulo m_i

Let's consider the operator of the following sort:

$$y^{(i)} = \left| \sum_{j=1}^{s} X_j \right|_{m_i},$$

where for numbers $X_1, X_2, \mathbf{K}, X_s$ is satisfied condition

$$0 \le \sum_{j=1}^{s} X_j < M = m_1 m_2 \, \mathbf{K} \, m_k$$

and

$$X_1 = b_{v-1,1}2^{v-1} + b_{v-2,1}2^{v-2} + \mathbf{K} + b_{1,1}2 + b_{0,1},$$
$$X_2 = b_{v-1,2}2^{v-1} + b_{v-2,2}2^{v-2} + \mathbf{K} + b_{1,2}2 + b_{0,2},$$
$$\mathbb{M}$$
$$X_{s-1} = b_{v-1,s-1}2^{v-1} + b_{v-2,s-1}2^{v-2} + \mathbf{K} + b_{1,s-1}2 + b_{0,s-1},$$
$$X_s = b_{v-1,s}2^{v-1} + b_{v-2,s}2^{v-2} + \mathbf{K} + b_{1,s}2 + b_{0,s},$$

$b_{r,j} \in \{0,1\}, \quad r = 0, 1, \mathbf{K}, v-1; \; j = 1, 2, \mathbf{K}, s.$

Algorithm 3.1.

Step 1. Are realized v of count operations of units modulo m_i:

$$b_{v-1}^{(i)} = \left| \sum_{j=1}^{s} b_{v-1,j} \right|_{m_i}, \quad \mathbf{K}, \quad b_1^{(i)} = \left| \sum_{j=1}^{s} b_1^{(i)} \right|_{m_i}, \quad b_0^{(i)} = \left| \sum_{j=1}^{s} b_0^{(i)} \right|_{m_i}.$$

Step. 2. Realizing modular products of values $b_r^{(i)}$ $(r = 0, 1, \mathbf{K}, v-1)$ on $\left|2^r\right|_{m_i}$ $(r = 0, 1, \mathbf{K}, v-1)$:

$$\mathbb{B}_{v-1}^{(i)} = \left| b_{v-1}^{(i)} \left| 2^{v-1} \right|_{m_i} \right|_{m_i},$$
$$\mathbb{M}$$
$$\mathbb{B}_1^{(i)} = \left| b_1^{(i)} |2|_{m_i} \right|_{m_i},$$
$$\mathbb{B}_0^{(i)} = b_0^{(i)}.$$

Step 3. Consider, that

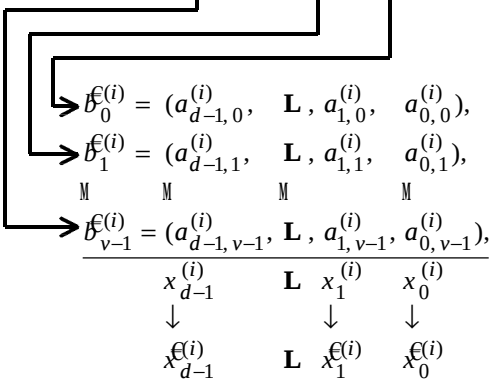
$$\mathbb{B}_0^{(i)} = x_1^{(i)}, \mathbb{B}_1^{(i)} = x_2^{(i)}, \mathbf{K}, \mathbb{B}_{v-2}^{(i)} = x_{n-1}^{(i)}, \mathbb{B}_{v-1}^{(i)} = x_n^{(i)}$$

and fulfil algorithm 2.1.

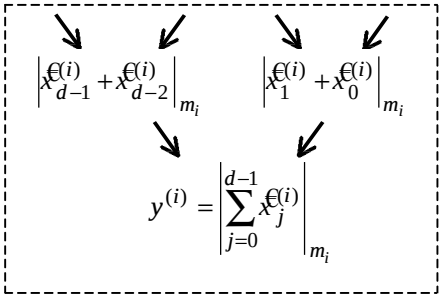
3.2. "Vertical" *N*-ary the converter of the code of a binary number system in the code modulo m_i

The structure of algorithm 3.1 can be presented as follows:

$$
\begin{array}{l}
X_1 = (b_{v-1,1}, \ldots, b_{1,1}, b_{0,1}),\\
X_2 = (b_{v-1,2}, \ldots, b_{1,2}, b_{0,2}),\\
\ldots\quad\ldots\quad\ldots\quad\ldots\\
X_s = (b_{v-1,s}, \ldots, b_{1,s}, b_{0,s}),\\
\hline
\quad b_{v-1}^{(i)} \ldots b_1^{(i)}\ b_0^{(i)}\\
\quad\downarrow\qquad\quad\downarrow\ \downarrow\\
\quad \tilde{b}_{v-1}^{(i)} \ldots \tilde{b}_1^{(i)}\ \tilde{b}_0^{(i)}
\end{array}
$$



$$
\begin{array}{l}
\tilde{b}_0^{(i)} = (a_{d-1,0}^{(i)}, \ldots, a_{1,0}^{(i)}, a_{0,0}^{(i)}),\\
\tilde{b}_1^{(i)} = (a_{d-1,1}^{(i)}, \ldots, a_{1,1}^{(i)}, a_{0,1}^{(i)}),\\
\ldots\quad\ldots\quad\ldots\quad\ldots\\
\tilde{b}_{v-1}^{(i)} = (a_{d-1,v-1}^{(i)}, \ldots, a_{1,v-1}^{(i)}, a_{0,v-1}^{(i)}),\\
\hline
\quad x_{d-1}^{(i)} \ldots x_1^{(i)}\ x_0^{(i)}\\
\quad\downarrow\qquad\quad\downarrow\ \downarrow\\
\quad \tilde{x}_{d-1}^{(i)} \ldots \tilde{x}_1^{(i)}\ \tilde{x}_0^{(i)}
\end{array}
$$



$$\left|\tilde{x}_{d-1}^{(i)} + \tilde{x}_{d-2}^{(i)}\right|_{m_i} \qquad \left|\tilde{x}_1^{(i)} + \tilde{x}_0^{(i)}\right|_{m_i}$$

$$y^{(i)} = \left|\sum_{j=0}^{d-1} \tilde{x}_j^{(i)}\right|_{m_i}$$

The block diagram of a parallel summing converter on a figure 6 is represented. The algorithm of operation of a converter of a sequential type is defined by the formula:

$$x^{(i)} = \left| b_{v-1}^{(i)} 2^{v-1} + b_{v-2}^{(i)} 2^{v-2} + \mathbf{K} + b_1^{(i)} 2 + b_0^{(i)} \right|_{m_i} =$$

$$= \left| \left| \left(\mathbf{K} \left| \left| \left(\left| b_{v-1}^{(i)} 2 \right|_{m_i} + b_{v-2}^{(i)} \right) 2 \right|_{m_i} + b_{v-3}^{(i)} \right) 2 \right|_{m_i} + \mathbf{K} + b_1^{(i)} \right) 2 \right|_{m_i} + b_0^{(i)} \Bigg|_{m_i} .$$

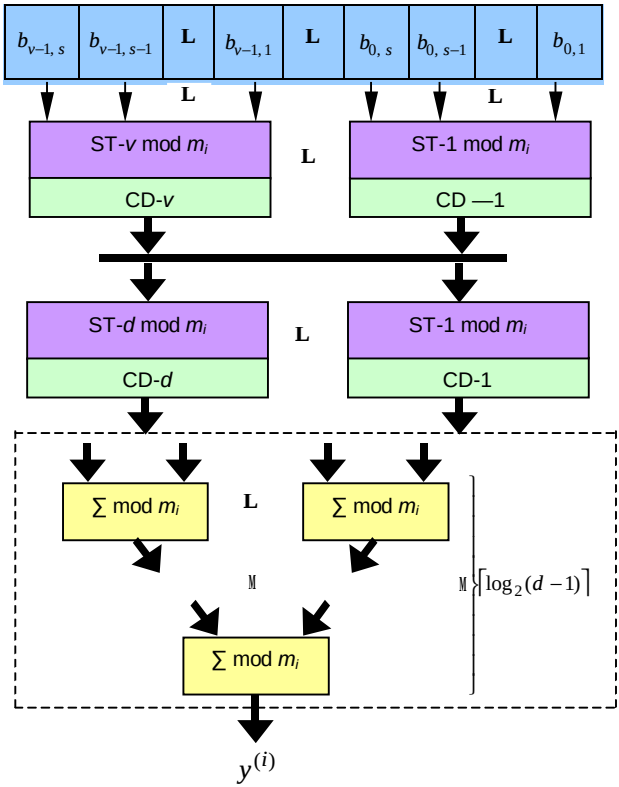


Figure 6. "Vertical" *N*-ary the converter of the code of a binary number system in the code modulo m_i

The schematic diagram of the *N*-ary converter of a sequential type will coincide with the scheme of the "vertical" summator of a sequential type (figure 4). But the sizes of an input of the serialparallel register of shift and amount of inputs of the counter of units ST-$\mathrm{mod}\, m_i$ will be

distinguished. An amount of clock ticks of dating pulses necessary for operation of a converter equal *v*.

4. Principle of *N*-ary multiplying modulo m_i

For implementation of *N*-ary multiplying modulo m_i it is possible to use property of a discrete log:

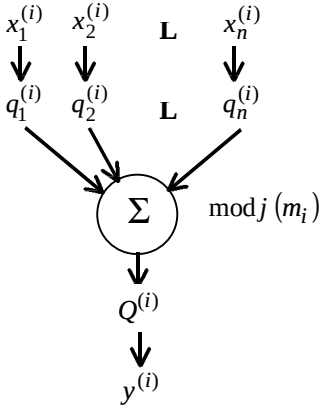
$$\left|\mathrm{Log}_g \prod_{j=1}^{n} x_j^{(i)}\right|_{j(m)} = \left|\sum_{j=1}^{n} \mathrm{Log}_g x_j^{(i)}\right|_{j(m)},$$

where $\mathrm{Log}_g A$ —discrete logarithm from A modulo m_i and basis g (g — primitive radical); $j(m_i)$ — function of Euler from value m_i;

$$j(m_i) = p_1^{a_1-1} p_2^{a_2-1} \quad \mathbf{K} \quad p_z^{a_z-1}(p_1-1)(p_2-1)\mathbf{K}(p_z-1),$$

where $p_1, p_2, \mathbf{K}, p_z$ —simple factors.

Then the structure of algorithm of *N*-ary multiplying modulo m_i can be presented as follows:



here $q_j^{(i)} = \mathrm{Log}_g x_j^{(i)}$, where $j = 1, 2, \mathbf{K}, n$; $Q^{(i)} = \left|\sum_{j=1}^{n} \mathrm{Log}_g x_j^{(i)}\right|_{j(m)}$;

$y^{(i)}$ outcome of calculation of an inverse discrete logarithm.

For implementation of *N*-ary multiplying modulo m_i the device, introduced on a figure 7 can be used. The device contains ROM-1.1 … ROM-1.*n*, intended for storage of values of a discrete logarithm; the *N*-ary summator modulo $j(m_i)$ or $P_{m_i}(m_i)$; ROM-2 intended for storage of outcomes of inverse discrete logarithm.

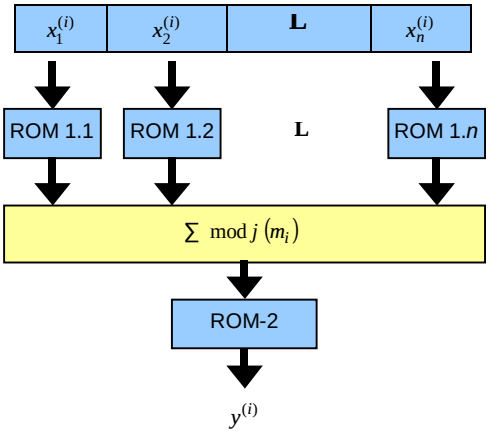


Figure 7. The *N*-ary multiplying tube modulo m_i

5. Conclusion

Thus outcomes obtained in this paper allowed to expand a circle of algorithmic and technical solutions for implementation of *N*-ary operations modulo m_i. The traditional devices use a so-called a "horizontal" method. The new algorithmic and technical solutions realize a "vertical" method. The advantage of a "vertical" method in comparison with a "horizontal" method at essential increase of an amount of operands —*N* is reached. The characteristics of binary arithmetic devices considerably depend on principles of construction of parallel counters of units modulo. The level of complexity of parallel counters is the linear function from number of entry arguments. Significant successes in the theory of synthesis of parallel counters recently are reached. The depth of the scheme was considerably reduced and the speed is raised.

The reached outcomes can become elements of the theory of *N*-ary finite ring computations.

6. References

1. F.J. Taylor, "Single Modulus ALU for Signal Processing", *IEEE Trans. Acoust., Speech, Signal processing*, vol. ASSP-33, N 5, October 1985, pp. 1302-1315.

2. M.A. Soderstrand, W.K. Jenkins, G.A. Jullien and F.T. Tailor, "*Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*", IEEE Press, New York, 1986.

3. H.J. Nussbaumer, "*Fast Fourier Transform and Convolution Algorithms*", Springer-Verlag, 1982.

4. J.M. Pollard, "The Fast Fourier Transform in a Finite Fild", *Math. Comp.*, vol. 25, N 114, 1971, pp.365-374.

5. D.E.R. Denning, "*Cryptography and Data Security*", Adisson-Wesly, Reading MA., 1983.

6. Oleg A. Finko, "Methods of Problem-Oriented Representation and Data Processing in Resources of the Hardware Support of Intellectual Systems", Proceedings of the IEEE Computer Society international Conference on Artificial Intelligence Systems 2002 (IEEE AIS'02). Gelendzhik, Russia, September 5-10, 2002, pp. 453-454.

7. Oleg A. Finko, "Sverkhparallel'nie Logicheskie Vichislenia Metodami Moduliarnoi Arifmetiki (Superparallel Logical Computations by Methods of Modular Arithmetics)", Trudi Mezhdunarodnikh Konfereztii "Iskusstvennie Intellektual'nie Sistemi" (IEEE AIS'02) i "Intellektual'nie SAPR" (CAD-2002), *Izdatel'stvo Fiziko-Matematicheskoi Literaturi*, Moskow, 2002. pp. 448-455.

8. Kai Hwang, "*Computer Arithmetic: Principles, Architecture and Design*'', John Wiley, 1979.

9. D. Zhang, G.A. Jullien, W.C. Miller, "VLSI Implementations of Neural-Like Networks for Finite Ring Computations", Proceedings of the 32nd Mid-West Symposium on Circuits and Systems, Champaign, III, August 14-16, 1989, vol. 1, New York (N. Y.), 1990, pp 485-488.

10. N.S. Szabo and R.I. Tanaka, "*Residue Arithmetic and its Applications to Computer Technology*", McGraw-Hill, New York, 1967.

11. Donald E. Knuth, "*Iskusstvo programmirovaniia, T. 2: Poluchislennye algoritmy*", Moscow: Vil'iams, 2000.

12. A.G. Akritas, "*Elements of Computer Algebra*", John Wiley & Sons, 1989.

13. Oleg A. Finko, "Number Restoration in the System of Residual Classes with a Minimum Number of Radices", *Engineering Simulation*, Overseas Publishers Association, Vol. 16, 1999, pp. 329-334.

14. Author's Certificate 1557682, MKI h 03 M 7/18, "Preobrazovatel' Pozitsionnogo Koda v Kod Sistemy Ostatochnykh Khlassov (Converter of a Positional code to the Code of the Residue Number Systems)" / O.A. Finko, V.A. Krasnobaeyv, and N.I. Shvetsov. *Bull*. No. 14, Published April 15, 1990. (In Russian).

15. Oleg A. Finko, "Check and Reconfiguration of Analog-to-Digital Devices Operating in the System of Residual Classes", *Engineering Simulation*, Overseas Publishers Association, Vol. 18, 2001, pp. 631-543.

16. Author's Certificate 1383365, MKI G 06 F 11/10, "Ustroistvo dlia Sviortki po Moduliu (The Device for Convolution Modulo)", / O.A. Finko, N.I. Cherviakov, N.I. Shvetsov, A.V. Palzhev, *Bull*. No. 11, Published March 23, 1988. (In Russian).

17. E. Earl, J.R. Swartzlander, "Parallel couters", *IEEE Trans. Comput*. vol. C-22, N 6, 1973, pp. 1021-1024.

18. S. Dormido, M.A. Canto, "An Upper Bound for the Syntesis of Generalized Parallel Couters", *IEEE Trans. Comput*. vol. C-34, N 8, 1982, pp. 802-805.

19. O.A. Muzichenko, "Ispol'zovanie Simmetrii Peremennikh Dlia Umen'shenia Slozhnosti Logicheskikh Skhem", *Avtomatika i Telemekhanika*, N 10, 2000, pp. 151-163.

20. O.A. Muzichenko, "Sintez Logicheskikh Skhem Modul'nogo Kontrolia v Unitarnikh Pozitsionnikh Dvoichnikh Kodakh", *Avtomatika i Telemekhanika*, N 3, 2001, pp. 159-173.