

**50 лет**

**модулярной  
арифметике**

**Юбилейная Международная  
научно-техническая конференция**

(В рамках V Международной  
научно-технической конференции  
«Электроника и информатика 2005»)





**Сборник научных трудов**


**Москва, Зеленоград, 2005**

Ваш С.А. Умрихин  
Ваш А. Корганов

 С.А. Умрихин

В.С. - Бузнецкий Р.Т.  
Александр / Бородин Р.Т.

 (М. Кузнецков)  
 / В. Шупаков

 (М.А. Орлов)

Иван / Корнев /  
Иван / Ю.С. Мезинин /  
Иван / А.А. Своякин /  
Иван / Евстигнев Р.А.

Министерство образования и науки Российской Федерации.

Федеральное агентство по образованию.

Московский государственный институт электронной техники  
(технический университет).

Открытое акционерное общество "Ангстрем".

Институт проблем проектирования в микроэлектронике (ИППМ) РАН.

Государственное унитарное предприятие "Научно-производственный  
центр СПУРТ".

Еженедельник PC-Week RE и Виртуальный компьютерный музей  
<http://www.computer-museum.ru>.

НИИ прикладных физических проблем им. А.Н.Севченко,  
г. Минск, Республика Беларусь.

Научно-внедренческая фирма "КРИПТОН", г. Киев, Украина.

Институт математики Национальной Академии Наук РК,  
Алма-Ата, Республика Казахстан.

Computer Algorithm & Program Development Corp., New York, USA.

---

Юбилейная Международная научно-техническая конференция

## **50 лет модулярной арифметике**

(В рамках V Международной научно-технической конференции  
"Электроника и информатика – 2005")

*(Зеленоград, 23 – 25 ноября 2005 г.)*

### **СБОРНИК НАУЧНЫХ ТРУДОВ**

ББК 32.973-04  
П99  
УДК 004.312 (066)

П99 «50 лет модулярной арифметике». Юбилейная Международная научно-техническая конференция (В рамках V Международной научно-технической конференции "Электроника и информатика – 2005"). Сборник научных трудов. – М.: ОАО «Ангстрем», МИЭТ, 2006, - 775 с.  
ISBN 5-7256-0409-8

В сборнике научных трудов Юбилейной конференции «50 лет модулярной арифметике» представлены результаты научных исследований в области Системы счисления остаточных классов (СОК) и модулярной арифметики на ее основе за пятидесятилетний период с момента зарождения СОК. В связи с намечающимся синтезом модулярной и троичной арифметик, представлены также материалы по троичной арифметике и трехзначной диалектической логике.

Организационный комитет:

Сопредседатели:

**Чаплыгин Ю.А., Амербаев В.М.**

Члены Оргкомитета:

**Бархоткин В.А., Стемпковский А.Л., Дьячков В.Н., Пак И.Т.,  
Чернявский А.Ф., William Khatskevitch,  
Пройдаков Э.М., Малашевич Б.М., Финько О.А.**

Тексты научных трудов представлены в авторской редакции

ISBN 5-7256-0409-8

© Ангстрем, 2006  
© МИЭТ, 2006



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

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

### **Abstract**

*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

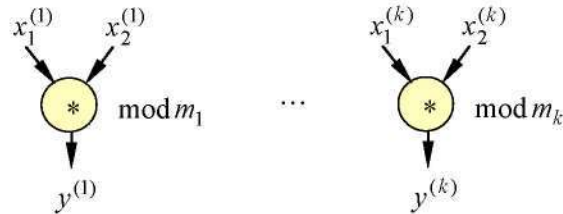
$$\{X\} = (x^{(1)}, x^{(2)}, \dots, x^{(k)})$$

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

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

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.

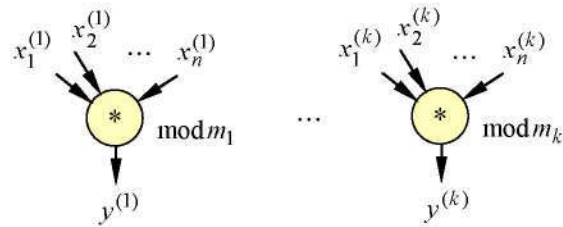
The existing algorithms MA use binary operations  $x^{(i)} = |x_1^{(i)} * x_2^{(i)}|_{m_i}$  ( $i = 1, 2, \dots, k$ ):



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

$$y^{(i)} = \left[ \begin{matrix} n \\ * \\ x_j^{(i)} \end{matrix} \right]_{m_i} :$$



The residuals  $x^{(1)}, x^{(2)}, \dots, x^{(k)}$  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:

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

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

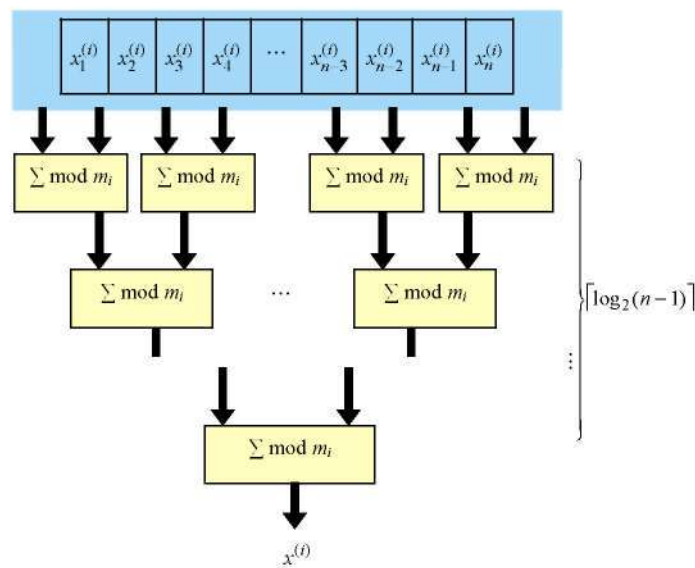
## 2. $N$ -ary operator of addition modulo $m_i$

### 2.1. Traditional architectures of $N$ -ary summaters 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  $T_{\Sigma} = \lceil \log_2(n-1) \rceil \tau_{\Sigma}$ , where  $\tau_{\Sigma}$  time of opera-



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

tion of one binary summator modulo  $m_i$ .

**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  $n$ .

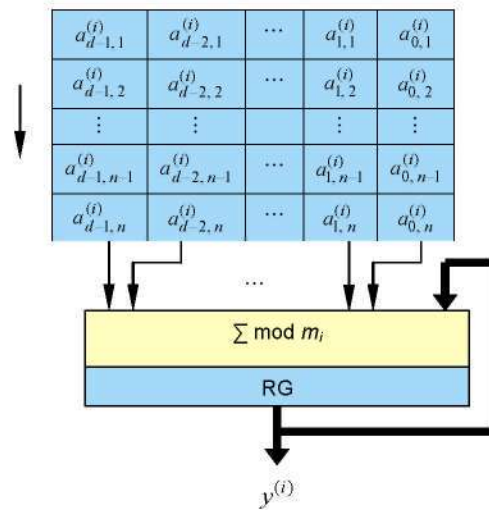


Figure 2. "Horizontal"  $N$ -ary summator modulo  $m_i$  of a sequential type

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

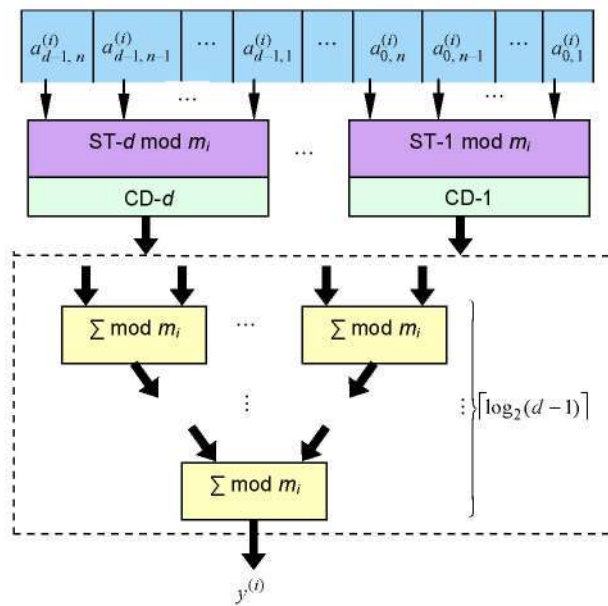
**2.2.1. The algorithm of "vertical"  $N$ -ary addition modulo  $m_i$ .** Let's consider algorithm  $N$ -ary addition modulo  $m_i$ , which works not with numbers  $x_j^{(i)}$  ( $j = 1, 2, \dots, n$ ), but with digits  $a_{r,1}^{(i)}, a_{r,2}^{(i)}, \dots, a_{r,n}^{(i)}$  ( $r = d-1, \dots, 1, 0$ ).

Algorithm 2.1.

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

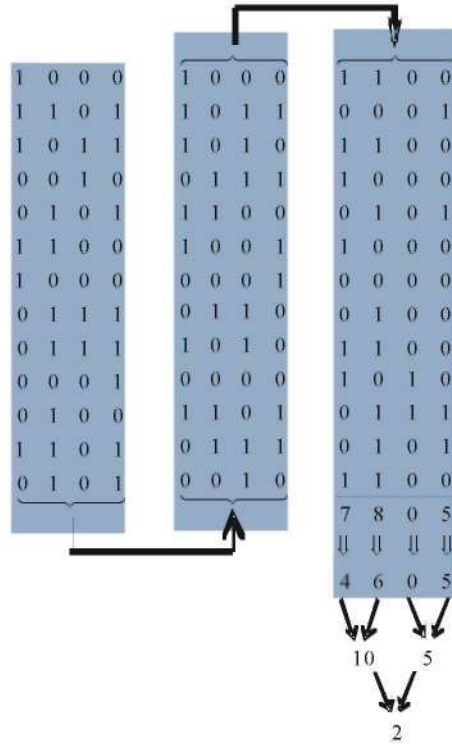


The device for parallel implementation of algorithm 2.1 (figure 3) contains the register of storage of binary bit digits,  $d$  of parallel counters of units (devices of convolution)  $ST \bmod m_i$  and  $d - 1$  of binary summators  $\Sigma \bmod m_i$ . If on outputs of parallel units counters the unitary code, the additional encoders  $CD-1 \dots CD-d$  of the unitary code in the code of a binary number system are installed. For the usually used module  $2 \leq m_i < 128$  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  $\xi_{d-1}^{(i)}, \dots, \xi_1^{(i)}, \xi_0^{(i)}$ , from algorithm 2.1 for construction of the formula (principle of Horner):

$$y^{(i)} = \left| \xi_{d-1}^{(i)} 2^{d-1} + \xi_{d-2}^{(i)} 2^{d-2} + \dots + \xi_1^{(i)} 2 + \xi_0^{(i)} \right|_{m_i} =$$

$$= \left| \left( \dots \left( \left( \left( \left( \xi_{d-1}^{(i)} 2 \right|_{m_i} + \xi_{d-2}^{(i)} \right) 2 \right|_{m_i} + \xi_{d-3}^{(i)} \right) 2 \right|_{m_i} + \dots + \xi_1^{(i)} \right) 2 \right|_{m_i} + \xi_0^{(i)} \right|_{m_i} .$$

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

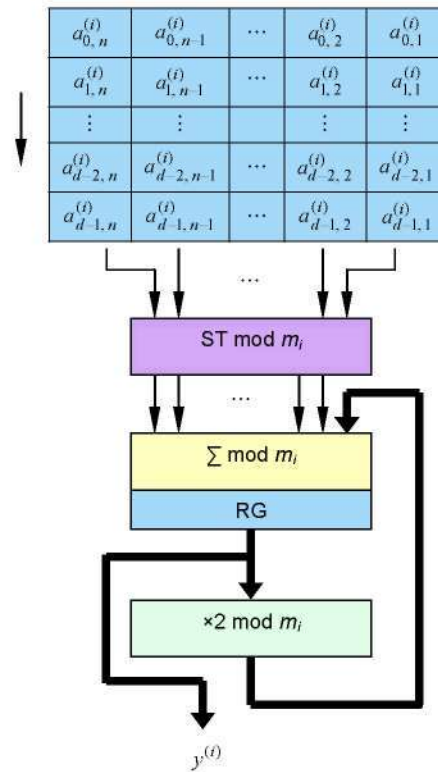
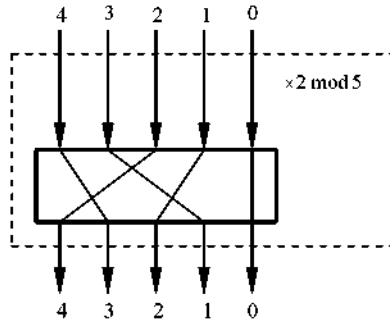


Figure 4. The "vertical"  $N$ -ary summator modulo  $m_i$  of a sequential type



**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  $\xi_0^{(i)}$ .

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  $X_1, X_2, \dots, X_s$  is satisfied condition

$$0 \leq \sum_{j=1}^s X_j < M = m_1 m_2 \dots m_k$$

and

$$\begin{aligned} X_1 &= b_{\nu-1,1} 2^{\nu-1} + b_{\nu-2,1} 2^{\nu-2} + \dots + b_{1,1} 2 + b_{0,1}, \\ X_2 &= b_{\nu-1,2} 2^{\nu-1} + b_{\nu-2,2} 2^{\nu-2} + \dots + b_{1,2} 2 + b_{0,2}, \\ &\vdots \\ X_{s-1} &= b_{\nu-1,s-1} 2^{\nu-1} + b_{\nu-2,s-1} 2^{\nu-2} + \dots + b_{1,s-1} 2 + b_{0,s-1}, \\ X_s &= b_{\nu-1,s} 2^{\nu-1} + b_{\nu-2,s} 2^{\nu-2} + \dots + b_{1,s} 2 + b_{0,s}, \end{aligned}$$

$$b_{r,j} \in \{0, 1\}, \quad r = 0, 1, \dots, \nu-1; \quad j = 1, 2, \dots, s.$$

Algorithm 3.1.

*Step 1.* Are realized  $\nu$  of count operations of units modulo  $m_i$ :

$$\beta_{\nu-1}^{(i)} = \left| \sum_{j=1}^s b_{\nu-1,j} \right|_{m_i}, \quad \dots, \quad \beta_1^{(i)} = \left| \sum_{j=1}^s b_{1,j} \right|_{m_i}, \quad \beta_0^{(i)} = \left| \sum_{j=1}^s b_{0,j} \right|_{m_i}.$$

*Step 2.* Realizing modular products of values  $\beta_r^{(i)}$  ( $r = 0, 1, \dots, \nu-1$ ) on

$$\left| 2^r \right|_{m_i} \quad (r = 0, 1, \dots, \nu-1):$$

$$\hat{\beta}_{\nu-1}^{(i)} = \left| \beta_{\nu-1}^{(i)} \left| 2^{\nu-1} \right|_{m_i} \right|_{m_i},$$

$\vdots$

$$\hat{\beta}_1^{(i)} = \left| \beta_1^{(i)} \left| 2 \right|_{m_i} \right|_{m_i},$$

$$\hat{\beta}_0^{(i)} = \beta_0^{(i)}.$$

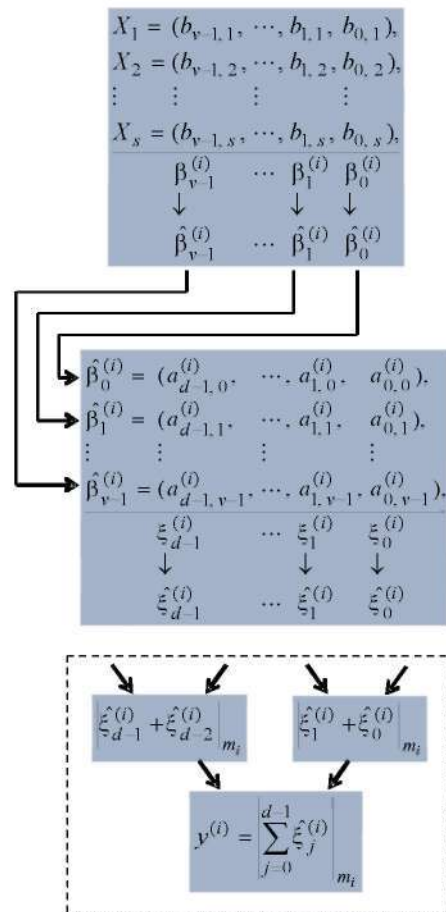
*Step 3.* Consider, that

$$\hat{\beta}_0^{(i)} = x_1^{(i)}, \hat{\beta}_1^{(i)} = x_2^{(i)}, \dots, \hat{\beta}_{\nu-2}^{(i)} = x_{n-1}^{(i)}, \hat{\beta}_{\nu-1}^{(i)} = x_n^{(i)}$$

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:



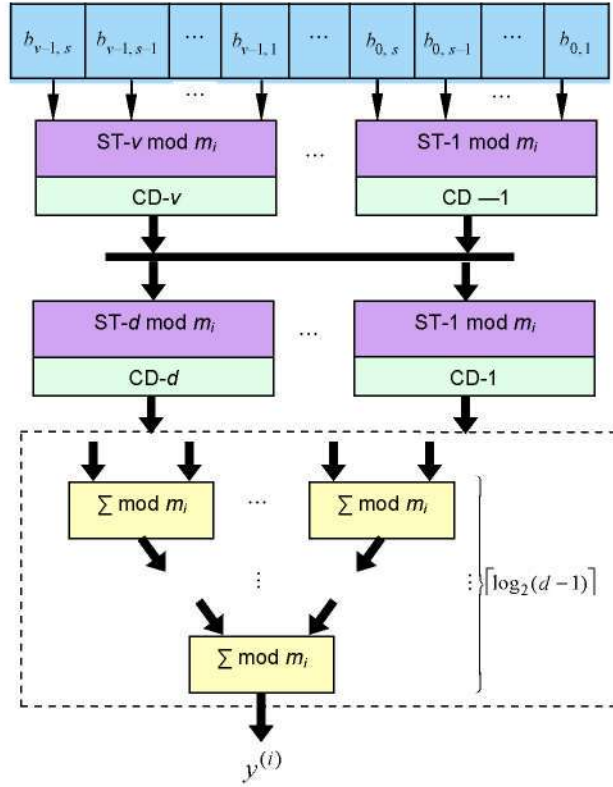


The block diagram of a parallel summing converter on a figure 6 is represented. The algorithm of operation of a converter of a sequential type is defined by the formula:

$$x^{(i)} = \left| \beta_{v-1}^{(i)} 2^{v-1} + \beta_{v-2}^{(i)} 2^{v-2} + \dots + \beta_1^{(i)} 2 + \beta_0^{(i)} \right|_{m_i} =$$

$$= \left| \left( \dots \left| \left( \left| \beta_{v-1}^{(i)} 2 \right|_{m_i} + \beta_{v-2}^{(i)} \right) 2 \right|_{m_i} + \beta_{v-3}^{(i)} \right) 2 \right|_{m_i} + \dots + \beta_1^{(i)} \right) 2 \right|_{m_i} + \beta_0^{(i)} \right|_{m_i}$$

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-mod $m_i$  will be distinguished. An amount of clock ticks of dating pulses necessary for operation of a converter equal  $v$ .



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

#### 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:

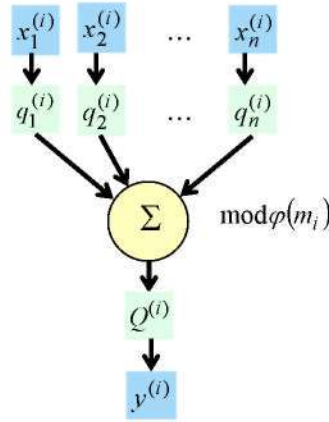
$$\left| \text{Log}_g \prod_{j=1}^n x_j^{(i)} \right|_{\varphi(m)} = \left| \sum_{j=1}^n \text{Log}_g x_j^{(i)} \right|_{\varphi(m)},$$

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

$$\varphi(m_i) = p_1^{\alpha_1-1} p_2^{\alpha_2-1} \dots p_z^{\alpha_z-1} (p_1-1)(p_2-1)\dots(p_z-1),$$

where  $p_1, p_2, \dots, p_z$  —simple factors.

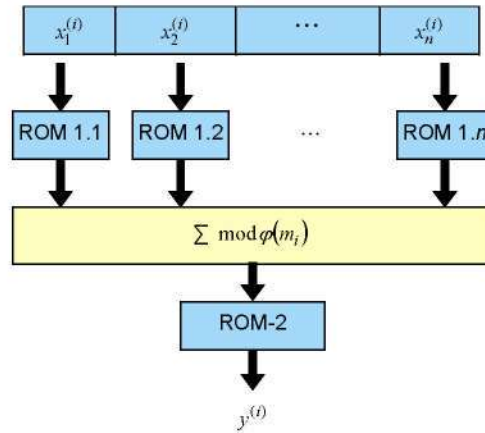
Then the structure of algorithm of  $N$ -ary multiplying modulo  $m_i$  can be presented as follows:



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

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  $\varphi(m_i)$  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. Taylor, "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 Field", *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. O.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. O.A. Finko, "Sverkhparallel'nie Logicheskie Vichislenia Metodami Moduliamoi 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 32<sup>nd</sup> 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.

