The implementation of L-codes in the system of residual classes

A. Yanko
ORCID: 0000-0003-2876-9316
Poltava National Technical Yuri Kondratyuk University, Poltava, Ukraine

V. Krasnobayev
ORCID: 0000-0001-5192-9918
V. N. Karazin Kharkiv National University, Kharkiv, Ukraine

A. Martynenko
ORCID: 0000-0002-9576-0138
Poltava National Technical Yuri Kondratyuk University, Poltava, Ukraine

Introduction

Currently, the possibilities of R-codes for error correction in the system of residual classes (SRC) are being intensively investigated. This is due to the simplicity of the structure of R-codes and good corrective capabilities, as well as the comparative simplicity of their construction for any given minimum code distance.

It is important and interesting to consider the so-called linear codes (L-codes) in the SRC. In the literature, L-codes are described qualitatively rather than quantitatively. The fact is that, to date, no one has studied the properties of systems of residual classes, whose bases are not mutually prime numbers [1]. Such a system also has certain corrective properties, which makes it necessary to assess the possibility and feasibility of using such systems to improve the reliability of computer systems and components. Therefore, this important and promising issue is considered in this article.

Keywords: computer systems and components of fast processing of integer data; correction codes; error correction; greatest common divisor; lowest common multiple; system of residual classes.
the reliability of computer systems and components of fast processing of integer data (CSCPID).

The sum, difference, and product of any vectors of a linear code are code words. In this case, non-code words cannot be associated with any natural numbers [2]. We show that error correction in the SRC with the help of L-codes leads to hardware redundancy equivalent to reservation. To this end, we consider two known theorems.

**Theorem 1.** The minimum distance of the correction L-code in the system of residual classes is equal to the minimum weight of nonzero code words. From the theorem it follows that the minimum code distance can be determined if weights of code words are known.

**Theorem 2.** In order for a L-code to have a minimum distance $d_{min}$, it is necessary and sufficient that the degree of redundancy satisfies the relation:

$$R = M^{d_{min}}.$$

From Theorem 2, it follows that the correction of arbitrary errors of information in the SRC with the help of L-codes leads to a large redundancy equivalent to reservation. Thus, it is ineffective to use linear codes for error correction, which with equal probability correspond to arbitrary distortions of the code words residuals in the SRC [3]. However, if we restrict the class of possible errors in the individual code word residues, the possibilities of the L-codes are significantly extended.

Consider lemma 1. For any integer $A = (a_1, a_2, ..., a_n)$ in the system of residual classes with bases $m_i$ $(i = 1, n)$ and for any pair of bases $m_i$ and $m_j$, the following condition must be true:

$$(a_i - a_j) \equiv 0 \pmod{d_{ij}},$$

where $d_{ij}(m_i, m_j)$ is the greatest common divisor of bases $m_i$ and $m_j$, and $i, j = 1, n; i \neq j$.

To determine the necessary and enough conditions for the detection of one-time errors using the L-codes according to the results of lemma 1, the following theorem was formulated and proved.

**Theorem 3.** To detect errors in the remainder of an arbitrary base $m_i$ $(i = 1, n)$ of a number $A = (a_1, a_2, ..., a_n)$, specified in the system of residual classes with bases $m_1, ..., m_n$, it is necessary that the base $m_i$ has at least one, different from one, common divisor with the other bases $m_j$ $(i \neq j)$ [4].

Proof. Let the greatest common divisor (GCD) $d_{ij}(m_i, m_j)$ be defined for arbitrary SRC bases $(i \neq j)$, and the error occurred at the base $m_i$, i.e. $a_i = a_i + \Delta a_i$. We show that the expression $(a_i - a_j) \pmod{d_{ij}}$ is equivalent to $\Delta a_i \pmod{d_{ij}}$. According to the lemma, the following equality holds:

$$(a_i - a_j) \equiv 0 \pmod{d_{ij}}.$$

Write the expression:

$$a_i + \Delta a_i = a_i \pmod{m_i}$$

as

$$a_i + \Delta a_i = m \cdot m_i + a_i,$$

where $m$ is an integer.

From the last expression we define the distorted remainder:

$$a_i - a_j = [(a_i - a_j) + (-md_{ij}) + \Delta a_i].$$

Because:

$$(a_i - a_j) \equiv 0 \pmod{d_{ij}}$$

and $-md_{ij} \equiv 0 \pmod{d_{ij}}$,

or $md_{ij} = kd_{ij}$, and $k$ is a natural number, then:

$$\Delta a_i \equiv (md_{ij}).$$

Obviously, in the absence of common dividers, i.e. if $d_{ij} = 1$, then $\Delta a_i \equiv (md_{ij})$. This proves the necessary condition of the theorem.

The necessary condition of the theorem is sufficient if the error is not a multiple of the divisor $d_{ij}$.

In fact, if

$$(md_{ij} + a_{iy}) \neq 0 \pmod{d_{ij}},$$

For $0 < a_{iy} < d_{ij}$.

Theorem 3 can be formulated as follows. To detect an error in the residual at an arbitrary base $m_i$ of the number $A = (a_1, a_2, ..., a_n)$, specified in the SRC, it is necessary and sufficient that the error $\Delta a_i$ is not a multiple of the divisors $d_{ij}$ and $d_i = (d_{i1}, d_{i2}, ..., d_{in})$.

where $d_i$ is the GCD of the divisors $d_i = (d_{i1}, d_{i2}, ..., d_{in})$.

Based on the results of Theorem 3, we construct an error detection algorithm.

1. Check the residual by the base $m_i$. To do this, we define a set of values:

$$a_1 - a_2 = a_{12} \pmod{d_{12}},$$

$$a_1 - a_3 = a_{13} \pmod{d_{13}},$$

$$a_1 - a_n = a_{1n} \pmod{d_{1n}}.$$

If $a_{iy} = (md_{ij})$, then the second residue is checked, and so on.

2. To obtain the values $a_{iy} (i \neq j)$ we compile a matrix:
When compiling the matrix $G$ it is not necessary to indicate the true numerical value $a_{ij}$, it is enough to present its distinguishing feature:

$$a_{ij} = \begin{cases} 0, & \text{if } a_i - a_j = 0 \pmod{d_{ij}}, \\ 1, & \text{if } a_i - a_j \neq 0 \pmod{d_{ij}}. \end{cases}$$

3. If the determinant of the matrix $|G| = 0$, then the number $A = (a_1, a_2, \ldots, a_n)$ is correct, and if $|G| \neq 0$, then the number $A$ is incorrect. Now for the considerations that simplify the above algorithm.

Based on the fact that:

$$a_i - a_j = (d_{ij} - (a_i - a_j)) \pmod{d_{ij}},$$

we can skip the step of finding determinant $|G|$. It is enough to define the diagonal elements of the matrix $G$ and add one value $a_{ni}$, i.e.

$$a_{12}, a_{23}, a_{34}, \ldots, a_{n-1n}, a_{nn}.$$  

It is easy to check that with such values of $a_{ij}$, it is possible to establish not only the fact of code word distortion, but also to determine the number of the distorted possible values of base errors $m_i$, i.e. multiples of the divisor $d_{ik} - K_{d_{ik}}$ and multiples of the divisor $d_{ik} - K_{d_{ik}}$. Thus, the number of possible values of detectable errors is:

$$m_i - 1 - (K_{d_{ik}} + K_{d_{ij}} - K_{d_{ik}, d_{ij}}).$$

To ensure compliance with the possible values of the errors on the basis $m_i$ it is necessary to fulfill inequality (1).

Q.E.D.

The necessary condition of Theorem 4 is sufficient if different values of the error values $\Delta a_i$ correspond to different product values $a_{ik} \cdot a_{ij}$ and vice versa. Indeed, in this case there is a one-to-one correspondence between the possible values $\Delta a_i$ and the values of the product $a_{ik} \cdot a_{ij}$, which determines the possibility of uniquely determining the magnitude of the error [4].

Based on Theorem 4, we compose an error correction algorithm for an arbitrary base $m_i$:

1. Determine the number of distorted residual. To do this, we calculate the values:

$$a_1 - a_2 = a_{12} \pmod{d_{12}},$$

$$a_2 - a_3 = a_{23} \pmod{d_{23}},$$

$$\ldots$$

$$a_{n-1} - a_n = a_{n-1n} \pmod{d_{n-1n}},$$

$$a_n - a_1 = a_{n2} \pmod{d_{n2}}.$$  

If all residuals are $a_{ij} = 0 \pmod{d_{ij}}$, then the number $A$ is correct. If the error occurred at the base $m_i$, then $a_{ij} \neq 0$ and $a_{ik} \neq 0$. The number of different combinations of $a_{ij}, a_{ik}$ is $(d_{ij} - 1) \cdot (d_{ik} - 1)$, where $(d_{ij} - 1)$ is the number of possible values of $a_{ij}$ ($a_{ij} \neq 0$), $(d_{ik} - 1)$ is the number of possible values of $a_{ik}$ ($a_{ik} = 0$), and the number of possible values of base errors $m_i$ is $m_i - 1$ ($\Delta a_i \neq 0$) minus the number of undetected errors. The number of undetected errors consists of the number of errors, multiples of the divisor $d_{ik} - K_{d_{ik}}$ and multiples of the divisor $d_{ik} - K_{d_{ik}}$. Thus, the number of possible values of detectable errors is:

$$m_i - 1 - (K_{d_{ik}} + K_{d_{ij}} - K_{d_{ik}, d_{ij}}).$$

2. By the values of $a_{ij}$ and $a_{ik}$ appeal to the block of error constants, where we select the appropriate value of $\Delta a_i$.

3. We perform the correction of the number $A$ in and we get the correct number $A = A - \Delta A$, i.e.

$$A = (a_1, a_2, \ldots, a_i, \ldots, a_n).$$
If in the abbreviated SRC due to the exclusion of the base on which the error occurred, it is possible to unambiguously represent a number \( A \), then instead of determining by the values of \( a_{ij} \) and \( a_{ik} \) the value of the error \( \Delta a_i \), we will directly calculate the values of the correct remainder \( a_i \).

Consider this error correction algorithm:
1. Calculate the value of residuals \( a_{12}, a_{23}, \ldots, a_{n_1} \).
2. Determine the number of distorted balances. Let the error occurred at the base \( m_j \). In this case, this basis is excluded, and the number \( A \) is presented on the bases \( m_1, m_2, \ldots, m_n \), i.e.
\[
A = (a_1, a_2, \ldots, a_{n-1}, a_{n+1}, \ldots, a_n).
\]
3. Perform a convolution of the number \( A \) into positional code.
4. Determine the true value of the distorted residual:
\[
a_i = A - \left[ A / m_j \right] m_j,
\]
where \( \lfloor x \rfloor \) is the whole part \( x \), not exceeding \( x \).

Corrected number:
\[
A_{\text{cor}} = (a_1, a_2, \ldots, a_{i-1}, a_i, \ldots, a_n).
\]

Let us determine the conditions under which it is possible to exclude some bases from the SRC. To do this, we present the bases of the original SRC in the canonical form:
\[
m_1 = \beta_{11}^{a_{11}} \beta_{12}^{a_{12}} \cdots \beta_{1i}^{a_{1i}},
m_2 = \beta_{21}^{a_{21}} \beta_{22}^{a_{22}} \cdots \beta_{2i}^{a_{2i}},
\ldots
\]
\[
m_n = \beta_{n1}^{a_{n1}} \beta_{n2}^{a_{n2}} \cdots \beta_{ni}^{a_{ni}},
M = \beta_1^{a_1} \beta_2^{a_2} \cdots \beta_i^{a_i}.
\]

To uniquely determine the number \( A \), specified in the SRC with bases \( m_1, m_2, \ldots, m_n \), and lying in the range \([0, M)\) it is possible to exclude only those bases for which \( \beta_{m_i} = \beta_{li} \), \( (m = \bar{1}, k, \ i = \bar{1}, n) \). It is necessary that
\[
a_{m_i} \geq a_{li}.
\]

Thus, the necessary and sufficient conditions for error correction are determined by eliminating the distorted base. These conditions are the simultaneous fulfillment of equality and inequality:
\[
\beta_{m_i} = \beta_{li}, \ a_{m_i} \geq a_{li}.
\]

Let the SRC be given by bases \( m_1 = 4, \ m_2 = 6, \ m_3 = 12, \ m_4 = 18 \). Wherein \( M = [4, 6, 12, 18] = 36 \).

In accordance with the possibility of the condition of error correction (2), we will determine those bases of the SRC that can be excluded.

Imagine the base of SRC in the canonical form: \( m_1 = 2^2 \), \( m_2 = 2 \cdot 3 \), \( m_3 = 2^3 \cdot 3 \), \( m_4 = 2 \cdot 3^2 \) and \( M = 2^3 \cdot 3^2 \).

Obviously, the sought bases are \( m_2, m_3, m_4 \). Let’s make a check, for which we compose the particular values of the LCM:
\[
M_1 = [6, 12, 18] = 36,
M_2 = [4, 12, 18] = 36,
M_3 = [4, 6, 18] = 36,
M = [4, 6, 12] = 36.
\]

The value of the LCM is \( M_4 < 36 \), which confirms the correctness of the definition of excluded grounds from a given SRC.

Above, the algorithm for detecting and correcting errors in the SRC by means of \( L \)-codes was described. Let be \( (a_k - a_{i+1}) \mod d_{k+1} \) when calculating values, it is determined that \( a_{i+1} \neq 0, \ a_{i+1} \neq 0, \) and all other values are:
\[
a_{k+1} = (a_k - a_{i+1}) \mod d_{k+1} = 0.
\]

Then it is stated that the number \( A \) is incorrect, and the error is present in the remainder of the base \( m_i \), i.e.
\[
A = (a_1, a_2, \ldots, a_i, \ldots, a_n).
\]

Referring by the values \( a_{i+1} \) and \( a_{i+1} \) to the block of error constants, we determine the error value \( \Delta a_i \) and then we determine the true value of the residual:
\[
a_{i,\text{cor}} = a_i - \Delta a_i.
\]

The corrected number will appear as:
\[
A_{\text{cor}} = (a_1, a_2, \ldots, a_{cor}, \ldots, a_n).
\]

To correct an error with the help of the developed correction method, it is necessary that the error \( \Delta a_i \) is not at the same time divisible by two dividers \( d_{i-1} \) and \( d_{i+1} \), which limits the class of corrected errors [4].

Thus, there is a need to develop effective methods and algorithms to expand the class of possible correctable errors.

The method of correction of one-time errors, allowing to correct errors that are multiples of one of the dividers \( d_{i-1} \) or \( d_{i+1} \), is as follows.

Let a SRC be set with mutually not simple bases, i.e. \( \gcd (m_1, m_2, \ldots, m_n) \geq 2 \).

And let a number be given in the SRC:
\[
A_{\text{cor}} = (a_1, a_2, \ldots, a_n).
\]

We define all values \( a_{k+1} \), i.e. \( a_{12}, a_{23}, a_{34}, \ldots, a_{n-1} \), \( a_n \). Without breaking the generality of reasoning, we assume that \( a_{i+1} \neq 0 \), and all other values are \( a_{k+1} \neq 0 \). Because:
\[ a_{i+1} = (a_i - a_{i-1}) \mod d_{i+1} \neq 0, \]

error may be present only in residues on the bases \( m_i \) or \( m_{i+1} \). In this regard, two hypotheses are possible:

- an error is present in the residual \( a_i \);
- an error is present in the residual \( a_{i+1} \).

Before we consider the error correction process by the proposed method, we formulate and prove a theorem, the result of which we use in determining the convergence process for the totality of numbers of the form:

\[ A^{(k)} = (a_1, \ldots, a_{i-1}, a_k, a_{i+1}, \ldots, a_n) \]

to the correct number:

\[ A^{(\rho)} = (a_1, \ldots, a_{i-1}, a_{i+1}, a_{i+2}, \ldots, a_n) \]

First consider the lemma.

Lemma 2. The sum, difference, and product of any \( L \)-code word are also code words.

**Theorem 5.** Let in the ordered \((m_i < m_j; i = 1, n)\) system of residual classes with bases \( m_1, m_2, \ldots, m_n\) an incorrect (distorted in one residue) number be given:

\[ A = (a_1, a_2, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n) \]

and let:

\[ \Delta a_i = a_i - a_{i-1} = k_i d_{i+1}. \]

Then in the set of values:

\[ a_{d_i} = (a_i - k_i d_{i+1}) \mod m_i \]

there is a single value of \( a_{d_i} \), at which the number:

\[ A^{(\rho)} = (a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n) \]

is the correct number, where \( d_{i+1} = \gcd(m_{i+1}, m_i) \), and \( k_i \) may take values \( k_i = 1, 2, ..., m_i / (d_{i+1} - 1) \).

Proof. We show that there is such a value of \( a_{d_i} \), at which the number:

\[ A = (a_1, a_2, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n) \]

is the correct number. By the condition of the theorem, the error \( \Delta a_i \) is a multiple of the divisor \( d_{i+1} \). The expression \( k_i d_{i+1} \) contains all possible multiples of \( d_{i+1} \).

Thus, there will be at least one value of \( k_i = \rho_i \), at which:

\[ \Delta a_{\rho_i} = \rho_i d_{i+1}, \]

and

\[ a_{\rho_i} = a_i - \Delta a_{\rho_i}. \]

We show that \( A^{(\rho)} \) is the only correct number from the set of numbers of the form \( A^{(k)} \).

Suppose there is such a value:

\[ a_{i_{\rho_2}} = a_i - \rho_2 d_{i+1}. \]

At which the number \( A^{(\rho_2)} \) is also correct. Then, in accordance with lemma 2, the number:

\[ A^{(\rho_3)} = (0, \ldots, a_{i_{\rho_1}} - a_{i_{\rho_2}}, \ldots, 0) \]

is correct. If the number \( A^{(\rho_1)} - A^{(\rho_2)} \) is correct, then in accordance with lemma 1 we have:

\[ (\rho_2 - \rho_1) d_{i+1} \equiv 0 \pmod{d_{i+1}} \]

\[ (\rho_2 - \rho_1) d_{i+1} \equiv 0 \pmod{d_{i-1}} \]

\[ (\rho_2 - \rho_1) d_{i+1} \equiv 0 \pmod{d_{i+1}} \]

If \( i \neq n \), then the only correct number \( A^{(\rho_1)} - A^{(\rho_2)} \) is the zero code word. This is due to the fact that \( d_{i+1} \neq 0 \) and \( d_{i+1} \) is not equal to the GCD of the dividers \( d_{i+1}, d_{i+2}, \ldots, d_n \).

Moreover, inequality \( d_{i+1} \neq [d_{i+1}, d_{i+2}, \ldots, d_n] \) contradicts the condition of arbitrary choice of bases \( m_1, m_2, \ldots, m_n \). Therefore, the following equality holds:

\[ A^{(\rho_1)} - A^{(\rho_2)} = (0, 0, \ldots, 0, \ldots, 0) \]

Thus, \( \rho_1 = \rho_2 \), that confirms the uniqueness of existence \( \rho_1 \), at which:

\[ A^{(\rho)} = (a_1, a_2, \ldots, a_{i_{\rho_1}}, \ldots, a_n) \]

is correct. Q.E.D.

We develop an error correction algorithm based on the result of Theorem 5.

Consider the first hypothesis. Since \( a_{i_{\rho_1}} = 0 \), the error is a multiple of the divisor \( d_{i+1} \). Therefore, an error on the basis can take values:

\[ \Delta a_i = k d_{i+1} \]

for \( k_i = 1, 2, \ldots, m_i / d_{i+1} - 1 \).

Calculate a set of values:

\[ a_{d_{i}} = (a_i - k_i d_{i+1}) \mod m_i \]

If in this set there is such a value of \( a_{d_{i}} \), at which:

\[ A^{(m)} = (a_1, a_2, \ldots, a_{i_{d_{i}}}, \ldots, a_n) \]

is the correct number, then the first hypothesis is valid, i.e. the error is present in the residual of the base \( m_i \). In this case, the corrected number is:

\[ A_{cor} = A^{(m)} \]

where:

\[ a_{d_{i}} = (a_i - m_i d_{i+1}) \mod m_i \]

If for all values of \( a_{d_{i}} \) the number \( A^{(k)} \) is incorrect,
then the value \( a_i \) is true, and the error occurred in the residual of the base \( m_{i+1} \). Since \( a_{i+1,i+2} = 0 \), then the error on the base \( m_{i+1} \) is a multiple of the divisor \( d_{i+1,i+2} \) i.e.

\[
\Delta a_{i+1} = k_{i+1} d_{i+1,i+2},
\]

where \( k_{i+1} = 1, 2, \ldots, m_{i+1}/d_{i+1,i+2} - 1 \).

Define a set of values:

\[
a_{i+1,k_{i+1}} = (a_{i+1} - k_{i+1} d_{i+1,i+2}) \mod m_{i+1}.
\]

According to theorem 5, in this set there is necessarily a single number \( a_{i+1,N} \), at which

\[A^{(N)} = (a_1, a_2, \ldots, a_{i+1}, \ldots, a_n) \]

is the correct number.

Note that the order of hypothesis testing is arbitrary and does not affect the probability of error correction.

However, in order to increase the speed of determining the number of a distorted residue, it is first necessary to test the hypothesis for which the value \( m_i/d_{k-1,k} \) (\( k = i, i + 1 \)) will be the lowest.

Consider an example of the implementation of the developed error correction algorithm using \( L \)-codes.

Let the SRC be given by bases \( m_1 = 4, m_2 = 6, m_3 = 12, m_4 = 18 \). Wherein \( M = 36, d_{12} = 2, d_{23} = 6, d_{34} = 6, d_{41} = 2 \). The amount of code words is presented in Table 1.

It is necessary to determine the correctness of the number \( A = (3, 5, 7, 7) \) and correct it in case of distortion.

1. We define the values \( a_{12} = 0, a_{23} = 2, a_{34} = 0, a_{41} = 0 \). Since \( a_{23} \neq 0 \), then the number \( A \) is wrong, and the error occurred in the second or in the third residual.

2. Since \( m_2/d_{12} > m_3/d_{34} \), the first hypothesis is that the error is assumed in the residual of the base \( m_2 \).

3. Calculate the values \( a_{3k_3} = a_3 - k_3 d_{23} \) for \( k_3 = 1 \).

Get \( a_{3k_3} = a_3 - k_3 d_{23} = 7 - 1 \cdot 6 = 1 \). The resulting number \( A^{(1)} = (3, 5, 1, 7) \) is not a code word (see Table 1), i.e. the first hypothesis is not true. An error occurred in the residual of the base \( m_2 \).

4. We correct the number \( A \). For this, by the values \( k_3 = 1, 2 \) we define the desired value \( a_{2k_3} = a_2 - k_2 d_{21} \):

\[
k_2 = 1, a_{2k_2} = a_2 - k_2 d_{21} = 5 - 1 \cdot 2 = 3,
\]

\[
k_2 = 2, a_{2k_2} = a_2 - k_2 d_{21} = 5 - 2 \cdot 2 = 1.
\]

Thus, we get two code words: \( A^{(1)} = (3, 3, 7, 7) \) and

\[A^{(2)} = (3, 1, 7, 7).
\]

From table 1 it can be seen that the only correct code word is the value \( A^{(2)} \), i.e. \( A_{\text{cor}} = A^{(2)} = (3, 1, 7, 7) \).

Thus, the developed method of error correction in the SRC allows to extend the class of corrected errors. This greatly expands the corrective possibilities of the \( L \)-codes in the class of deductions [4].

Consider the operation of the device for detecting errors using \( L \)-codes, in accordance with the above algorithm. This device contains the input register, modulo adders \( m_i \) and \( d_{li} \) (\( i = 2, n \)) and \( n - 1 \) - the input element OR (Fig. 1).

The algorithm of operation of this device corresponds to the error detection algorithm developed above [5].
Let the SRC be given by the bases $m_1 = 4$, $a_{23} = 2$.

$m_1 = 12$. Wherein:

$$\prod_{i=1}^{3} m_i = 288, \ L = M = [4, 6, 12] = 12, \ d_{12} = 2.$$  

$$d_{12} = 4.$$  

Determine the correctness of the number:

$$A = ((11), (001), (0111)).$$

At the output of the modulo adder $m_2$ we get:

$$\overline{a_2} = m_2 - a_2 = 0101,$$

at the output of the adder modulo $m_3 - a_3 = m_3 - a_3 = 0101$. At the output of the adder modulo $d_{12}$ we get:

$$(a_i + a_{12}) = 0(\mod d_{12}),$$

at the output of the adder $d_{13}$:

$$(a_i + a_{13}) = 0(\mod d_{13}).$$

At the output of the device there is no signal, i.e. the number $A$ is correct (see Table 1).

Let the number $A$ be distorted by the base $m_2$ and let

$$\Delta a_2 = 011,$$

$$A = ((0011), (0100), (0111)).$$

At the output of the modulo adder $m_2$ we get the number:

$$\overline{a_2} = m_2 - a_2 = 010,$$

and at the output of the modulo adder $m_3$ we get the number:

$$\overline{a_3} = m_3 - a_3 = 0101.$$

At the output of the modulo adder $d_{12}$ we get the number:

$$a_i + a_{12} = 1(\mod d_{12}) ,$$

and at the output of the modulo adder $d_{13}$:

$$a_i + a_{13} = 0(\mod d_{13}).$$

At the output of the device, we get operand 0001, i.e. the number is incorrect.

<table>
<thead>
<tr>
<th>$A_i$</th>
<th>Code words</th>
<th>$A$ in SRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>00 0000</td>
<td>0000</td>
</tr>
<tr>
<td>0001</td>
<td>01 001</td>
<td>0001</td>
</tr>
<tr>
<td>0010</td>
<td>10 010</td>
<td>0010</td>
</tr>
<tr>
<td>0011</td>
<td>11 011</td>
<td>0011</td>
</tr>
<tr>
<td>0100</td>
<td>00 100</td>
<td>0100</td>
</tr>
<tr>
<td>0101</td>
<td>01 101</td>
<td>0101</td>
</tr>
<tr>
<td>0110</td>
<td>10 010</td>
<td>0110</td>
</tr>
<tr>
<td>0111</td>
<td>11 011</td>
<td>0111</td>
</tr>
<tr>
<td>1000</td>
<td>00 010</td>
<td>1000</td>
</tr>
<tr>
<td>1001</td>
<td>01 011</td>
<td>1001</td>
</tr>
<tr>
<td>1010</td>
<td>10 100</td>
<td>1010</td>
</tr>
<tr>
<td>0101</td>
<td>11 101</td>
<td>1011</td>
</tr>
<tr>
<td>0111</td>
<td>11 101</td>
<td>1011</td>
</tr>
</tbody>
</table>
As can be seen from the considered examples of the specific performance of the error correction operation, using the L-codes, the error detection process is implemented extremely simply. The time of error detection for the SRC given by any system of bases is always equal to three conditional ticks and does not depend (as is observed for R-codes) on the number \( N \) of information bases [6].

We present some considerations that will simplify the above device for detecting errors (Fig. 2).

Fig. 2. Device for detecting errors

First, we prove the relation
\[
(a_i + \Delta a_i) = (a_i + a_i) \mod d_i, 
\]
on the basis of which we compose an error correction algorithm. Let the remainder \( m_j \) in the operand \( A = (a_1, a_2, \ldots, a_n) \) be distorted, i.e.
\[
a_j = (a_j + \Delta a_j) \mod m_j. 
\]

We write the system of equalities:
\[
k_1 = a_i - a_j = a_i + (m_j - a_j) = (a_i - a_j + m_j - \Delta a_j). 
\]
\[
k_2 = a_j - a_i = a_j + \Delta a_j - a_i = (a_j - a_i + a_i). 
\]

We add these equalities and get:
\[
k_1 + k_2 = m_j \mod d_j 
\]
or
\[
k_1 + k_2 = 0 \mod d_j. 
\]

Thus, it is shown that:
\[
(a_i + \Delta a_i) = (a_i + a_i) \mod d_i, 
\]
that is, in a device for error detection, instead of \( n - 1 \) modulo adders \( m_j \), it is sufficient to have only one modulo adder \( m_j \).

The developed algorithm for the implementation of the error detection process is determined by the following relations:
\[
a_2 + m_1 - a_1 = (a_2 + a_1) \mod d_{12}, 
\]
\[
a_3 + m_1 - a_1 = (a_3 + a_1) \mod d_{13}. 
\]

The above-discussed variants of devices for detecting errors in the SRC make it possible to guarantee the detection of a number \( A \), distortion, however, this does not determine the number of the base on which the residue was distorted [7].

Consider determining the number of the remainder by which the distortion of the number \( A \) occurred.

Let the SRC be given by the bases
\[
m_1 = 4, \quad m_2 = 6, 
\]
\[
m_3 = 12, \quad m_4 = 18. 
\]

Wherein
\[
L = M = [4, 6, 12, 18] = 36, 
\]
\[
d_{12} = 2, \quad d_{23} = 6, 
\]
\[
d_{34} = 6, \quad d_{41} = 2, \quad A = (0, 2, 8, 2). 
\]

Let the number \( A \) be distorted by the base \( m_4 \), i.e.
\[
a_4 = (a_4 - \Delta a_4) \mod m_4, 
\]
and let \( \Delta a_4 = 5 \).

At the output of the modulo adder \( m_2 \) we get the value \( a_2 = m_2 - a_2 = 4 \) ; at the output of the modulo adder \( m_3 \)
we get \(-\bar{a}_5 = m_4 - a_5 = 4\), at the output of the modulo adder \(m_4 - a_4 = m_4 - a_4 = 11\). At the output of the modulo adder \(d_{12}\) we get \((a_1 + a_2) \equiv 0(\text{mod } d_{12})\), at the output of the modulo adder \(d_{23}\) - \((a_4 + a_4) \equiv 0(\text{mod } d_{23})\), at the output of the modulo adder \(d_{34}\) - \((a_4 + a_4) \equiv 0(\text{mod } d_{34})\), at the output of the modulo adder \(d_{41}\) - \((a_4 + a_4) \equiv 1(\text{mod } d_{41})\).

At the inputs of the modulo adders \(d_{34}\) and \(d_{41}\) and there is a non-zero result of the operation \((a_4 + a_4) \text{ mod } d_{12}\), therefore the fourth element AND is open, i.e. a signal is present on the fourth output bus. It follows that the error occurred in the fourth base \(a_4\) (Table 3).

On the basis of the proved Theorem 4, the necessary condition for detecting errors in the modulo \(m_i\) residue is condition (Table 3). This condition is also sufficient if the error \(\Delta a_j = a_j - a_j\) is not a multiple of the divisors \(d_{i-1}\), i.e. the following two dividers \(d_{i-1}^{(i-1)} = (d_{i-1}^{(i-1)}, \Delta a_i) = 1\), \(d_{i-1}^{(i-1)} = (d_{i-1}^{(i-1)}, \Delta a_i) = 1\).

In accordance with the results of Theorem 4, we construct an error correction algorithm for an arbitrary basis \(m_i\):

1. Define all possible values of type:
   \((a_i - a_{ni}) =ke_i, i_{i}(\text{mod } d_{i+1})\),
   \((a_i - a_{i1}) =ke_i, i_{i}(\text{mod } d_{i+1})\),
   \((a_i - a_2) =ke_2, i_{i}(\text{mod } d_{i+1})\),
   \((a_i - a_3) =ke_3, i_{i}(\text{mod } d_{i+1})\),
   \((a_1 - a_1) =d_{1-n, i}(\text{mod } d_{1-n, i})\),
   \((a_i - a_1) =d_{i-n, i}(\text{mod } d_{1-n, i})\).

2. If all values (3) are equal to zero, then there is either no error or it is a multiple of each of the divisors \(d_{i-1}, d_{i+1}\), (a single error is assumed).

3. If \(a_{i-1} \neq 0, a_{i+1} \neq 0\), and all other values \(a_{ij} = 0\), then the error occurred in the module \(m_i\), i.e. \(a_i = a_i + \Delta a_i\) (\(1 \leq \Delta a_i \leq m_i - 1\)).

In accordance with the proven Theorem 3, the necessary condition for correcting the error in the residual \(a_j\) is condition (4), written in the general form:

\[
(d_{ik} - 1)(d_{ij} - 1) \geq \delta(\Delta a_i),
\]

where:

\[
\delta(\Delta a_i) = m_i - 1 - (K_{d_{ik}} + K_{d_{ij}} - K_{d_{ik}d_{ij}}),
\]

\(K_{d_{ik}}\) is the number of possible error \(\Delta a_i\) divisors by the base \(m_i\) (i.e. the number of possible divisors of the number \(m_i - 1\), of the value \(d_{ik}\);

\(K_{d_{ij}}\) is the number of possible error \(\Delta a_i\) divisors by the base \(m_i\), multiples of the value \(d_{ij}\).

The condition (4) is also sufficient if different pairs of values \(a_{ik}\) and \(a_{ij}\) correspond to different possible values \(\delta(\Delta a_i)\) of the base errors \(m_i\) (i =, n).

Consider an example of a specific implementation of the operation of error correction in the SRC, given by the bases \(m_1 = 4, m_2 = 6, m_3 = 12\). In this case, the table of code words \(L = [4, 6, 12] = 12\) is represented as tab. 3. Note that \(d_{i2} = (4, 6) = 2, d_{i3} = (6, 12) = 6, d_{i3} = (4, 12) = 4; \delta(\Delta a_i) = 2\) (Tab. 4), \(\delta(\Delta a_i) = 3\) (Tab. 5), \(\delta(\Delta a_i) = 8\) (Tab. 6), where

\[
\delta(\Delta a_i) = m_i - 1 - (K_{d_{i2}} + K_{d_{i3}} - K_{d_{i2}d_{i3}}),
\]

\[
\delta(\Delta a_i) = m_i - 1 - (K_{d_{i2}} + K_{d_{i3}} - K_{d_{i2}d_{i3}}),
\]

\[
\delta(\Delta a_i) = m_i - 1 - (K_{d_{i2}} + K_{d_{i3}} - K_{d_{i2}d_{i3}}).\]

Let it be necessary to determine the correctness of the number \(A = (11, 100, 0111)\). The initial number \(A\) is entered in the first and second input registers [8]. The first adder of the first group determines the value \(\bar{a}_1 = m_1 - a_1 = 01\), the second - \(\bar{a}_2 = m_2 - a_2 = 010\), and the third - \(\bar{a}_3 = m_3 - a_3 = 00101\). The first modulo adder \(a_{i2}\) determines the value \(a_{i2} = (a_1 + a_2) \text{ mod } 12\), the second - \(a_{i3} = (a_2 + a_3) \text{ mod } 13\), the third - \(a_{i3} = (a_3 + a_4) \text{ mod } 13\). Thus, from the outputs of the corresponding decoders, only the second switch receives the values \(a_{i2} = 1, a_{i3} = 3\), according to which (see Table 6) it determines the value of the modulo \(m_2\) error to be
inverted, i.e. $\Delta a_2 = 3$, which through the second decoder in binary code is fed to the first input of the second adder, the second input of which receives the value $a_2 = a_2 + \Delta a_2 = 100$. The adder of the second group determines the result of the operation:

$$ (\Delta a_2 + a_2) \mod m_2 = (m_2 - \Delta a_2 + a_2 + \Delta a_2) \mod m_2 = 001. $$

### Table 3 - Table of code words for a set of bases of the SRC

<table>
<thead>
<tr>
<th>A</th>
<th>Code words in SRC</th>
<th>A</th>
<th>Code words in SRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_1$</td>
<td>$m_2$</td>
<td>$m_3$</td>
<td>$m_4$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

The input of the device receives the corrected number (see Table 2) $A = (11, 001, 0111)$.

### Table 4 - Solutions table

<table>
<thead>
<tr>
<th>$a_{31}$</th>
<th>$a_{12} = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\Delta a_1 = 1$</td>
</tr>
<tr>
<td>2</td>
<td>$\Delta a_1 = 3$</td>
</tr>
</tbody>
</table>

### Table 5 - Solutions table

<table>
<thead>
<tr>
<th>$a_{23}$</th>
<th>$a_{12} = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\Delta a_2 = 5$</td>
</tr>
<tr>
<td>2</td>
<td>$\Delta a_2 = 3$</td>
</tr>
<tr>
<td>3</td>
<td>$\Delta a_2 = 1$</td>
</tr>
</tbody>
</table>

### Table 6 - Solutions table

<table>
<thead>
<tr>
<th>$a_{31}$</th>
<th>$a_{23}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>$\Delta a_1 = 7$</td>
</tr>
<tr>
<td>2</td>
<td>$\Delta a_1 = 2$</td>
</tr>
<tr>
<td>3</td>
<td>$\Delta a_1 = 1$</td>
</tr>
</tbody>
</table>

Thus, the error correction algorithms in the SRC with mutual in pairs non-simple bases make it relatively easy to implement a procedure for detecting and correcting one-time errors [4]. The considered scheme of detecting and correcting one-time errors makes it possible to localize the erroneous base and correct the error in one residual in just five conventional time ticks for any number of the SRC bases. The main advantages of the L-codes in the SRC is the simplicity of the procedure for detecting the location of an error and its localization. By the simplicity of decoding schemes, the L-codes have no analogues, both in the PNS and in the SRC [9].

### References

Engineering and Modern Technologies (FarEastCon), Vladivostok, 1-5.


