Efficient implementation of error correction codes in modular code

Nikolay Kucherov[1,2†], **Viktor Kuchukov**[1,2‡‡], **Elena Golimblevskaia**[1,2‡], **Natalia Kuchukova**[1††], **Irina Vashchenko**[1†‡] **and Ekaterina Kuchukova**[1,‡†]

[1] North-Caucasus Center for Mathematical Research, North-Caucasus Federal University, 1, Pushkin Street, 355017, Stavropol, Russia

[2] Sirius University of Science and Technology, 1, Olympic Ave, 354340, Sochi, Russia

E-mail: [†]nkucherov@ncfu.ru, [‡‡]vkuchukov@ncfu.ru, [‡]elena.golimblevskaya@gmail.ru, [††]nkuchukova@ncfu.ru, [†‡]irishechka.26@mail.ru, [‡†]ekuchukova@ncfu.ru

Abstract. The article develops an efficient implementation of an algorithm for detecting and correcting multivalued residual errors with a fixed number of calculations of the syndrome, regardless of the set of moduli size. Criteria for uniqueness are given that can be met by selecting moduli from a set of primes to satisfy the desired error correction capability. An extended version of the algorithm with an increase in the number of syndromes depending on the number of information moduli is proposed. It is proposed to remove the restriction imposed on the size of redundant moduli. Identifying the location of the error and finding the error vector requires only look-up tables and does not require arithmetic operations. In order to minimize the excess space, an extended algorithm is also proposed in which the number of syndromes and look-up tables increases with the number of information moduli, but the locations of errors can still be identified without requiring iterative computations. By using the approximate method, we have reduced the computational complexity of the algorithm for calculating the syndrome from quadratic to linear-logarithmic, depending on the number of bits in the dynamic range.

1. Introduction

Redundant residue number system (RRNS) error detection and correction has been used in several fault tolerant applications described in the literature [1]-[9]. In addition to correcting errors, a range overflow can be detected by adding two redundant moduli. RRNS has also been used to develop a fault-tolerant convolution algorithm that is well suited for implementation on multiprocessor systems [10, 8]. Using the long division method, the algorithm can detect and correct processing errors. In addition, the six-moduli RRNS based error correction code has been implemented on the hybrid storage [11, 7]. Compared to Reed-Solomon codes, this code provides a larger data storage with similar error correction capabilities [5, 6].

As a rule, the detection and correction of errors in RRNS is carried out in three sequential stages: checking for errors, identifying erroneous digits of the residue and correcting errors. Algorithms that fix only one residual digit error have been proposed in [12, 13, 14, 15]. In the case of using two redundant moduli, decoding of the value and location of a single error

can be solved using only one syndrome and one look-up table [15, 4]. On the other hand, detecting and correcting multiple residue digit errors is more difficult and time consuming. This is mainly due to the sheer number of combinations of different locations and residues to find errors in the second step. Typically, existing multiple error detection and correction algorithms use three different methods to locate the erroneous residue digits. This is a consistency check using the syndrome [16, 17], number recovery according to the Chinese Remainder Theorem (CRT) [18, 19] and the modulus projections [17, 20]. These algorithms require iterative and recursive computations and comparisons that defy parallel hardware implementation.

2. Projection method

Let's consider an approach to detecting and correcting multiple errors in RRNS [21, 3]. The digits of the RRNS residues are divided into three groups so that any combination of errors can be uniquely identified by one of the seven error location categories. From the obtained representation of the residue, three syndromes are deduced to detect up to $2t$ and correct up to t errors of the residue digits, where the number of redundant moduli is equal to $2t$. Regardless of the number of moduli, the error residue representation can be unambiguously extracted by these syndromes from six look-up tables in a few steps and subtracted directly from the resulting residue digits to correct the errors. The delay is fixed and does not depend on the size of the moduli set. The second criterion for the selection of modules, which imposes a large excess space, can be completely eliminated, making the number of calculations of the syndrome dependent on the number of information moduli. Compared to existing algorithms, the proposed algorithm is simpler and has a very low complexity of error decoding. Only one modulo subtraction is required for each syndrome computation. The proposed algorithm classifies various combinations of deduction errors into seven groups of locations without limiting the amount of information and redundant moduli, which is in stark contrast to the [15]algorithm, which simply finds one deduction error for RRNS with only two redundant moduli.

The first algorithm for detecting and correcting multiple residue digits errors was introduced in [16]. The error decoding algorithm establishes the order dependence in the residue digits in such a way that it is able to correct single packet errors up to t adjacent residue digits. The syndrome computation method calculates the difference between the received residue digits and the residue digits in extended bases. The $|\Delta|_{p_i}$ syndroms for $i = 1, 2, \ldots, k, k+1, \ldots, k+2t$ are grouped into the following sets:

$$\begin{aligned} S_1 &\equiv \left\{ |\Delta|_{p_{k+1}}, |\Delta|_{p_{k+2}}, \ldots, |\Delta|_{p_{k+2t}} \right\} \\ S_2 &\equiv \left\{ |\Delta|_{p_{k-t}}, |\Delta|_{p_{k-t+1}}, \ldots, |\Delta|_{p_{k+t-1}} \right\} \\ S_3 &\equiv \left\{ |\Delta|_{p_{k-2t-1}}, |\Delta|_{p_{k-2t}}, \ldots, |\Delta|_{p_{k-2}} \right\} \\ &\vdots \\ S_{[(k+t-1)/(t+1)]+1} &\equiv \left\{ |\Delta|_{p_1}, |\Delta|_{p_2}, \ldots, |\Delta|_{p_{2t}} \right\} \end{aligned} \tag{1}$$

where k is a number of information moduli.

The location of the error of the received digits of the residue is determined by S_i for $i = 1, 2, \ldots, [(k+t-1)/(t+1)] + 1$, provided that the number of nonzero elements is less than or equal to t. An error-free residue representation is then constructed by replacing the erroneous residue digits with the corresponding base extended residue digits. The complexity of this algorithm depends on the number of test sets and the number of syndromes calculated for each set.

Instead of looking for erroneous residue digits in [18] another algorithm was proposed using the continued fraction method. The correct fraction of error is then used to compute the value of the error-free residue representation. Calculations of fractions are solved by a recursive Euclidean

algorithm. However, this method requires more iterations for more moduli and residue digit errors, which makes this method ineffective for hardware implementation.

Sun and Krishna [17] introduced a coding theory approach to error control in RRNS and introduced the concepts of Hamming weight, minimum distance, weight distribution, and error detection and correction capabilities in RRNS. Four algorithms have been proposed, namely single error correction and multiple error detection, double error correction and multiple error detection, single batch error correction, and extended double-digit error correction. The first three algorithms require the computation of syndromes, as in [16], except that only excess residue digits are used in the computations. Erroneous digits of residues are found by matching of all digits of errors calculated by syndromes. Although the number of computed syndromes is less than in [16], the consistency check requires iterative computations involving every possible combination of error locations. The advanced double error correction algorithm uses a different approach from the first three algorithms. It uses the modulus projection concept introduced in [22] to correct residue singular errors, and is an extension of the residue singular error correction algorithm proposed in [23]. The location of erroneous residues is obtained when the value of the representation of a number after excluding two erroneous digits of the residue falls within the allowed range. The value is checked by calculating the mixed digits of the root of the representation of the residue of each projection. This process is similar to the first three algorithms in that it requires iterative estimation of every possible combination of residue digit error locations.

In [19] two algorithms based on CRT, unambiguous coding and list decoding, were proposed for detecting and correcting small and large numbers of residue digit errors, respectively. The unambiguous encoding method is an extended version of the algorithm proposed in [18]. It looks for two unknown integers y and z of bounded size, such that $|y \cdot \tilde{X}|_N \equiv z$, where $\tilde{X}$ is the resulting representation of the residue, and N is the dynamic range of the RRNS. The search for y and z must be calculated recursively until z/y becomes an integer. The list decoding method searches for a sequence of integers $c_0, c_1, \ldots, c_l$, of certain limited sizes from the received residue digits to form a polynomial $C(x) = \sum_{i=0}^{l} c_i x^i$ of degree l such that $\left|\sum_{i=0}^{l} c_i x^i\right|_N = 0$, where l is a function of the number of information and redundant moduli, as well as the values of the largest and smallest moduli. The error-free representation of the residue is restored by solving the polynomial for integer roots and finding their matching amplitudes with the resulting residue digits.

The error correction algorithm in [20] uses the same unit projection concept as the extended 2-digit error correction algorithm in [17], but it uses the CRT instead of the generalized weigted number system (GWNS) to compute the value of each projection. The algorithm is able to correct up to t errors in the size of the residues, ignoring the t digits of the residues from the obtained residues at each iteration of the calculation of the value. If the calculated value falls within the valid range, the ignored residue digits are invalid. The maximum number of iterations required in this algorithm is C_t^n, where n is the total number of moduli. [19] also proposed an extended iteration reduction scheme based on the maximum likelihood decoding (MLD) method. It first calculates the value of r of the residue digits, where r is the number of redundant residue digits. Only when the calculated value is within the valid range will the remaining residue digits be calculated using the radix expansion method. If t or fewer digits of the extended bases of the residue differ from the resulting digits of the residue, then the digits of the extended bases of the residue are unerring digits of the residue. The required number of iterations is $[(C_t^n)/(C_t^r)]$, which is significantly less than in the first method, especially when the number of modules is large.

In general, recursive or iterative computations and comparisons are required to determine erroneous residue digits for existing error detection and correction algorithms with multiple residue digits. Since the number of iterations depends on the location of bit errors, the time

taken to detect and correct bit errors is non-deterministic. The worst case increases with the size of the selected set of moduli, which depends on the number of information moduli and the error correction capabilities of the algorithms. Consequently, these algorithms are not amenable to hardware implementation.

3. Syndrome method

From the expression of the error vector

$$E_l = \left| \sum_{i=1}^{t} \frac{P_N}{p_{l_i}} \left| \left| P_{l_i}^{-1} \right|_{p_{l_i}} e_{l_i} \right|_{p_{l_i}} \right|_{P_N} = \sum_{i=1}^{t} a_{l_i} \frac{P_N}{p_{l_i}} - r_n P_N = a \frac{P_N}{\prod_{i=1}^{t} p_{l_i}} = a \prod_{j=1, l_j \neq l_i}^{n-t} p_{l_j} \tag{2}$$

E_l is a multiple of the product of error-free moduli. When $p_{k+2}, \ldots, p_{k+r} > p_i$, $\forall i \leq k$ and $t \leq [r/2]$, the minimum difference between the values of any two vectors of errors is always more than P_K. Beginning with $X < P_K$, a misrepresentation of the residue can be detected by checking if X' falls within the invalid range. However, decoding X' from $(x'_1, x'_2, \ldots, x'_n)$ using CRT requires large computational operations modulo P_N. Once errors have been found, the location of the erroneous residues must be located before they can be corrected. Typically, locating the erroneous residues requires time-consuming iterative computations that increase with the size of the selected set of moduli. To overcome the above disadvantages, this section proposes a new method for calculating the syndrome. It uses smaller modulo operations and the number of computations is fixed regardless of the size of the set of moduli.

3.1. Segregation of syndromes for identification of errors

The value of E_l is limited to P_N. Thus, each error vector corresponds to a unique E_l. Considering the value of the E_l error, it is possible to uniquely determine the corresponding error digits e_{l_i} and the location of the l_i error. Unfortunately, E_l cannot be obtained by directly decoding the representation of the residue $(x'_1, x'_2, \ldots, x'_n)$, since X is unknown.

To avoid the use of large error decoding operations modulo P_N, n residue digits $(x'_1, x'_2, \ldots, x'_n)$ X are divided into three different groups, $(x'_1, x'_2, \ldots, x'_n)$, $\left(x'_{k+1}, x'_{k+2}, \ldots, x'_{k+t}\right)$ and $\left(x'_{k+t+1}, x'_{k+t+2}, \ldots, x'_{k+r}\right)$, so that the modulo operations required to unambiguously resolve the error vector are about three times less than P_N.

The dynamic range P_N of RRNS $\left\{p_1, p_2, \ldots, p_k, p_{k+1}, \ldots, p_(k+r)\right\}$, where $p_{k+1}, p_{k+2}, \ldots, p_{k+r} > p_i$, $\forall i \leq k$ can be divided into

$$P_N = \prod_{i=1}^{k} p_i \times \prod_{i=k+1}^{k+r} p_i = P_K \times P_U \times P_V \tag{3}$$

where $P_K = \prod_{i=1}^{k} p_i$ is a product of information moduli, $P_U = \prod_{i=k+1}^{k+t} p_i$ is a product of the first t redundant moduli, and $P_V = \prod_{i=k+t+1}^{k+r} p_i$ is the product of the remaining t redundant moduli.

For convenience, we define the following notation:

X denotes the value of error-free representation of RRNS $(x_1, x_2, \ldots, x_k, x_{k+1}, \ldots, x_{k+r})$. X falls within the range $[0, P_{K-1}]$.

$\tilde{X}$ denotes the value of the resulting representation of the RRNS $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_k, \tilde{x}_{k+1}, \ldots, \tilde{x}_{k+r})$. $\tilde{X}$ falls within the range $[0, P_{N-1}]$.

$\tilde{X}_K$, $\tilde{X}_U$ and $\tilde{X}_V$ denote the values of the RNS representations for information moduli, for the first t redundant moduli, for the remaining t redundant moduli respectively, that is $\tilde{X}_K \equiv (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_k)$, $\tilde{X}_U \equiv (\tilde{x}_{k+1}, \tilde{x}_{k+2}, \ldots, \tilde{x}_{k+t})$ and $\tilde{X}_V \equiv (\tilde{x}_{k+t+1}, \tilde{x}_{k+t+2}, \ldots, \tilde{x}_{k+r})$. $\tilde{X}$, $\tilde{X}_K$, $\tilde{X}_U$ and $\tilde{X}_V$ may be erroneous or non-erroneous.

E_K, E_U and E_V denote the values of the error digits located in the information channels of the moduli, in the first t of the redundant channels of the moduli, and in the remaining t of the redundant channels of the moduli, that is, $E_K \equiv (e_{k_1}, e_{k_2}, \ldots, e_{k_i}, 0, \ldots, 0)$, $E_U \equiv \Big(0, \ldots, 0, e_{u_1}, e_{u_2}, \ldots, e_{u_j}, 0, \ldots, 0\Big)$ and $E_V \equiv (0, \ldots, 0, e_{v_1}, e_{v_2}, \ldots, e_{v_l})$, where $(k_1, k_2, \ldots, k_i)$, $(u_1, u_2, \ldots, u_j)$ and $(v_1, v_2, \ldots, v_l)$ represent the positions of the error digits and $i, j, l \leq t$. E_K, E_U and E_V are limited by P_N and can be computed from their respective error digits using the formula (2).

Based on the above notation, $\tilde{X}_K$, $\tilde{X}_U$ and $\tilde{X}_V$ can be expressed in terms of $\tilde{X}$ through

$$\tilde{X}_K = \left|\tilde{X}\right|_{P_K} \tag{4}$$

$$\tilde{X}_U = \left|\tilde{X}\right|_{P_U} \tag{5}$$

$$\tilde{X}_V = \left|\tilde{X}\right|_{P_V} \tag{6}$$

If there is no error, $\tilde{X} = X$. Based on the formulas (4) – (6),

$$\tilde{X}_K = |X|_{P_K} = X \tag{7}$$

$$\tilde{X}_U = |X|_{P_U} = X \tag{8}$$

$$\tilde{X}_V = |X|_{P_V} \tag{9}$$

6 It should be noted that (8) is valid if and only if $P_U > P_K$, and P_U and P_K are coprime.

Otherwise, if there is erroneous residue digit(s) in one or more groups among $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_k)$, $(\tilde{x}_{k+1}, \tilde{x}_{k+2}, \ldots, \tilde{x}_{k+t})$ and $(\tilde{x}_{k+t+1}, \tilde{x}_{k+t+2}, \ldots, \tilde{x}_{k+r})$, $\tilde{X}$ can be expressed as $\tilde{X} = X + E$, that is,

$$\tilde{X}_K = |X + E_K|_{P_K} = X + |E_K|_{P_K} - \alpha_K P_K \tag{10}$$

$$\tilde{X}_U = |X + E_U|_{P_U} = X + |E_U|_{P_U} - \alpha_U P_U \tag{11}$$

$$\tilde{X}_V = |X + E_V|_{P_V} = |X_V|_{P_V} + |E_V|_{P_V} - \alpha_V P_V \tag{12}$$

where

$$\alpha_K = \begin{cases} 0, & if \quad X + |E_K|_{P_K} < P_K \\ 1, & if \quad X + |E_K|_{P_K} \geq P_K \end{cases} \tag{13}$$

$$\alpha_U = \begin{cases} 0, & if \quad X + |E_U|_{P_U} < P_U \\ 1, & if \quad X + |E_U|_{P_U} \geq P_U \end{cases} \tag{14}$$

$$\alpha_V = \begin{cases} 0, & if \quad |X_V|_{P_V} + |E_V|_{P_V} < P_K \\ 1, & if \quad |X_V|_{P_V} + |E_V|_{P_V} \geq P_K \end{cases} \tag{15}$$

Since the resulting residue digits are grouped into three different groups, erroneous residue digits may exist in one of the following categories depending on their error location.

- EL1: erroneous digit(s) of the residue only in $\tilde{X}_K$.
- EL2: erroneous digit(s) of the residue only in $\tilde{X}_U$.
- EL3: erroneous digit(s) of the residue only in $\tilde{X}_V$.
- EL4: erroneous digits of the residue in $\tilde{X}_K$ and $\tilde{X}_U$.

- EL5: erroneous digits of the residue in $\tilde{X}_K$ and $\tilde{X}_V$.
- EL6: erroneous digits of the residue in $\tilde{X}_U$ and $\tilde{X}_V$.
- EL7: erroneous digits of the residue in $\tilde{X}_K$, $\tilde{X}_U$ and $\tilde{X}_V$.

In principle, the presence of any t remaining errors in the numbers of n residues can be detected by two syndromes, δ_1 and δ_2. δ_1 is calculated from $(x'_1, x'_2, \ldots, x'_n)$ and $\left(x'_{k+1}, x'_{k+2}, \ldots, x'_{k+t}\right)$, δ_2 is calculated from $(x'_1, x'_2, \ldots, x'_n)$ and $\left(x'_{k+t+1}, x'_{k+t+2}, \ldots, x'_{k+r}\right)$. Each of these syndrome computations includes the residue digits $(k+t)$. Since the dynamic ranges of δ_1 and δ_2 are much smaller than P_N, each syndrome value can be displayed in more than one error vector with no more than t remaining digits in different l_i locations and e_{l_i} values. Therefore, the two syndromes are not enough to correct the errors entered in the three different groups of residual digits. The third syndrome δ_3, calculated from $\left(x'_{k+1}, x'_{k+2}, \ldots, x'_{k+t}\right)$ and $\left(x'_{k+t+1}, x'_{k+t+2}, \ldots, x'_{k+r}\right)$, is required. Having found the common vector of errors, represented by three syndromes, it is possible to accurately determine the errors of the residue digit, provided that $P_U > P_K$.

Three syndroms δ_1, δ_2 and δ_3 can be calculated as follows:

$$\delta_1 = \left|\tilde{X}_U + \tilde{X}_K\right|_{P_U} \tag{16}$$

$$\delta_2 = \left|\tilde{X}_V + \tilde{X}_K\right|_{P_V} \tag{17}$$

$$\delta_3 = \left|\tilde{X}_V + \tilde{X}_U\right|_{P_V} \tag{18}$$

Replacing (7)-(9) with (16)-(18), we get

$$\delta_1 = |X - X|_{P_U} = 0 \tag{19}$$

$$\delta_2 = \left||X|_{P_V} - X\right|_{P_V} = \left|-\left\lfloor\frac{X}{P_V}\right\rfloor P_V\right|_{P_V} = 0 \tag{20}$$

$$\delta_3 = \left||X|_{P_V} - X\right|_{P_V} = 0 \tag{21}$$

Therefore, when there is no erroneous residue digit, the syndromes are $\delta_1 = \delta_2 = \delta_3 = 0$.

If any digit of the residue contains an error, then the values of the three syndromes can also be used to determine the category of the place of the error. Replacing (10)-(12) with (16)-(18), the values δ_1, δ_2 and δ_3 for each possible error location category can be calculated as follows:

EL1:

$$\delta_1 = \left|X - \left(X + |E_K|_{P_K} - \alpha_K P_K\right)\right|_{P_U} = \left|\alpha_K P_K - |E_K|_{P_K}\right|_{P_U} \tag{22}$$

$$\delta_2 = \left|X - \left(X + |E_K|_{P_V} - \alpha_K P_K\right)\right|_{P_U} = \left|\alpha_K P_K - |E_K|_{P_K}\right|_{P_V} \tag{23}$$

$$\delta_3 = \left||X|_{P_V} - X\right|_{P_V} = 0 \tag{24}$$

EL2:

$$\delta_1 = \Big|X - \Big(X + |E_U|_{P_U} - \alpha_U P_U\Big)\Big|_{P_U} = |E_U|_{P_U} \quad (25)$$

$$\delta_2 = \Big||X|_{P_V} - X\Big|_{P_V} = 0 \quad (26)$$

$$\delta_3 = \Big||X|_{P_V} - \Big(X + |E_U|_{P_U} - \alpha_U P_U\Big)\Big|_{P_V} = \Big|\alpha_U P_U - |E_U|_{P_U}\Big|_{P_U} \quad (27)$$

EL3:

$$\delta_1 = |X - X|_{P_U} = 0 \quad (28)$$

$$\delta_2 = \Big||X|_{P_V} - \Big(X + |E_V|_{P_V} - \alpha_V P_V\Big)\Big|_{P_V} = |E_V|_{P_V} \quad (29)$$

$$\delta_3 = \Big|\Big(|X|_{P_V} + |E_V|_{P_V} - \alpha_V P_V\Big) - X\Big|_{P_V} = |E_V|_{P_V} \quad (30)$$

EL4:

$$\begin{aligned}\delta_1 &= \Big|\Big(X + |E_U|_{P_U} - \alpha_U P_U\Big) - \Big(X + |E_K|_{P_K} - \alpha_K P_K\Big)\Big|_{P_U} = \\ &= \Big||E_U|_{P_U} - |E_K|_{P_K} + \alpha_K P_K\Big|_{P_U} \quad (31)\end{aligned}$$

$$\delta_2 = \Big||X|_{P_V} - \Big(X + |E_K|_{P_K} - \alpha_K P_K\Big)\Big|_{P_V} = \Big|\alpha_K P_K - |E_K|_{P_K}\Big|_{P_V} \quad (32)$$

$$\delta_3 = \Big||X|_{P_V} - \Big(X + |E_U|_{P_U} - \alpha_U P_U\Big)\Big|_{P_V} = \Big|\alpha_U P_U - |E_U|_{P_U}\Big|_{P_V} \quad (33)$$

EL5:

$$\delta_1 = \Big|X - \Big(X + |E_K|_{P_K} - \alpha_K P_K\Big)\Big|_{P_U} = \Big|\alpha_K P_K - |E_K|_{P_K}\Big|_{P_K} \quad (34)$$

$$\begin{aligned}\delta_2 &= \Big|\Big(|X|_{P_V} + |E_V|_{P_V} - \alpha_V P_V\Big) - \Big(X + |E_K|_{P_K} - \alpha_K P_K\Big)\Big|_{P_V} = \\ &= \Big||E_V|_{P_V} - |E_K|_{P_K} + \alpha_K P_K\Big|_{P_V} \quad (35)\end{aligned}$$

$$\delta_3 = \Big|\Big(|X|_{P_V} + |E_V|_{P_V} - \alpha_V P_V\Big) - X\Big|_{P_V} = |E_V|_{P_V} \quad (36)$$

EL6:

$$\delta_1 = \Big|\Big(X + |E_U|_{P_U} - \alpha_U P_U\Big) - X\Big|_{P_U} = |E_U|_{P_U} \quad (37)$$

$$\delta_2 = \Big||X|_{P_V} - \Big(X + |E_V|_{P_V} - \alpha_V P_V\Big)\Big|_{P_V} = |E_V|_{P_V} \quad (38)$$

$$\begin{aligned}\delta_3 &= \Big|\Big(|X|_{P_V} + |E_V|_{P_V} - \alpha_V P_V\Big) - \Big(X + |E_U|_{P_U} - \alpha_U P_U\Big)\Big|_{P_V} = \\ &= \Big||E_V|_{P_V} - |E_U|_{P_U} + \alpha_U P_U\Big|_{P_U} \quad (39)\end{aligned}$$

EL7:

$$\delta_1 = \Big|\Big(X + |E_U|_{P_U} - \alpha_U P_U\Big) - \Big(X + |E_K|_{P_K} - \alpha_K P_K\Big)\Big|_{P_U} =$$
$$= \Big||E_U|_{P_U} - |E_K|_{P_K} + \alpha_K P_K\Big|_{P_U} \quad (40)$$
$$\delta_2 = \Big|\Big(|X|_{P_V} + |E_V|_{P_V} - \alpha_V P_V\Big) - \Big(X + |E_K|_{P_K} - \alpha_K P_K\Big)\Big|_{P_V} =$$
$$= \Big||E_V|_{P_V} - |E_K|_{P_K} + \alpha_K P_K\Big|_{P_K} \quad (41)$$
$$\delta_3 = \Big|\Big(|X|_{P_V} + |E_V|_{P_V} - \alpha_V P_V\Big) - \Big(X + |E_U|_{P_U} - \alpha_U P_U\Big)\Big|_{P_V} =$$
$$= \Big||E_V|_{P_V} - |E_U|_{P_U} + \alpha_U P_U\Big|_{P_U} \quad (42)$$

3.2. Error detection and correction

From (22)-(42) there are nine different syndrome expressions, and each syndrome can take three of these nine expressions, as shown in table 1.

Table 1. Possible syndrome expressions for error location categories EL1-EL7

δ	Matching values	Error vectors E_K	E_U	E_V	Categories
δ_1	$v_1 = \Big\|\|E_U\|_{P_U} - \|E_K\|_{P_K} + \alpha_K P_K\Big\|_{P_U}$	+	+		EL4, EL7
	$v_2 = \Big\|\alpha_K P_K - \|E_K\|_{P_K}\Big\|_{P_U}$	+			EL1, EL5
	$v_3 = \|E_U\|_{P_U}$		+		EL2, EL6
δ_2	$v_4 = \Big\|\alpha_K P_K - \|E_K\|_{P_K}\Big\|_{P_V}$	+			EL1, EL4
	$v_5 = \Big\|\|E_V\|_{P_V} - \|E_K\|_{P_K} + \alpha_K P_K\Big\|_{P_V}$	+		+	EL5, EL7
	$v_6 = \|E_V\|_{P_V}$			+	EL3, EL6
δ_3	$v_7 = \Big\|\alpha_U P_U - \|E_U\|_{P_U}\Big\|_{P_U}$		+		EL2, EL4
	$v_8 = \|E_V\|_{P_V}$			+	EL3, EL5
	$v_9 = \Big\|\|E_V\|_{P_V} - \|E_U\|_{P_U} + \alpha_U P_U\Big\|_{P_V}$		+	+	EL6, EL7

Table 2 shows the mapping of error location categories to expressions of the three syndromes. Each syndrome value in the column must match an error vector in the column error location category, and each error vector in the error location category must also map to at least one of the three syndrome values in the column.

Six look-up tables are built from tables 1 and 2. In each look-up table, the values for the error vectors are precomputed and stored along with their error vectors. If the look-up key in the look-up table matches the look-up value, the error vectors that match the look-up value will be retrieved. Table 3 shows the types of (E_K, E_U, E_V) errors corresponding to error vectors stored in each look-up table, look-up values (computed from v_1 to v_9) and a search key (δ_1, δ_2 or δ_3), used to identify error location categories (EL1-EL7), and search for error vectors. For example, from the first row of the table 3 we can conclude that T_1 was created to extract the error vector E_K for EL1, EL4 and EL5. The search values v_2 and v_4 for T_1 are calculated for

Table 2. Mapping error location categories to syndromes

	EL1	EL2	EL3	EL4	EL5	EL6	EL7
δ_1	v_1	v_3	0	v_1	v_2	v_3	v_1
δ_2	v_4	0	v_6	v_4	v_5	v_6	v_5
δ_3	0	v_7	v_8	v_7	v_8	v_9	v_9

all possible combinations of $\alpha_K \in \{0, 1\}$ and E_K for t or less residue digit errors. If an error is identified as being in EL1 or EL5 (or EL1 or EL4), its error vector E_K can be extracted from T_1 using the δ_1 syndrome (or δ_2). The exact error vector is recovered in two stages. The error location category must be identified before the corresponding error vector extracted from the look-up table is used to correct the erroneous digits of the residues.

Table 3. Characteristics of look-up tables for searching error vectors

Table	Error vectors			Value substitutions	Key searching	Groups of location errors
	E_K	E_U	E_V			
T_1	+			v_2	δ_1	EL1, EL5
T_1	+			v_4	δ_2	EL1, EL5
T_2		+		v_3	δ_1	EL2, EL6
T_2		+		v_7	δ_3	EL2, EL4
T_3			+	v_6	δ_2	EL3, EL6
T_3			+	v_8	δ_3	EL3, EL5
T_4	+	+		v_1	δ_1	EL4, EL7
T_5	+		+	v_5	δ_2	EL5, EL7
T_6		+	+	v_9	δ_3	EL6, EL7

If only one of the three syndromes is zero, as indicated in table 2, the error location category can be identified by syndrome zero. This is EL1 for $\delta_3 = 0$, EL2 for $\delta_2 = 0$ and EL3 for $\delta_1 = 0$. The error vectors for EL1, EL2, and EL3 can be obtained from look-up tables T_1, T_2 and T_3 respectively, using either of the two nonzero syndromes. If none of the syndromes is zero, the error location category can be EL4, EL5, EL6, or EL7, as indicated in table 2. In this case, the actual category of the error location can be determined by checking the consistency of the error vectors received by δ_1, δ_2 and δ_3, according to the table 3. For example, since $E_{KV} = E_K \cup E_V$, if errors fall into EL5, the error vector E_{KV}, obtained from T_5 on δ_2, must include the error vectors E_K and E_V, obtained from T_1 and T_3 on δ_1 and δ_3 respectively. After locating the actual error location category, the erroneous residue digits are corrected by subtracting the resulting error vector from the resulting residue digits.

Let $E = \mathrm{find}\,(T, \delta)$ be the table look-up function that returns the error vector E for the look-up key δ into the look-up table T or zero if the corresponding record is not found. Steps required to detect and correct errors with multiple residue digits:

Step 1. Decode $\tilde{X}_K$, $\tilde{X}_U$ amd $\tilde{X}_V$ from the obtained residue digits.

Step 2. Calculate δ_1, δ_2 and δ_3 by (19), (20) and (21) respectively.

Step 3. If $\delta_1 = \delta_2 = \delta_3$, then there is no error, go to step 10.

Step 4. If only one of δ_1, δ_2 and δ_3 is not zero, then there are more than t residue digit errors. Go to step 10. .

Step 5. If $\delta_1 \neq 0$, $\delta_2 \neq 0$ and $\delta_3 = 0$, then the erroneous digit(s) of the residue are in the EL1. $E_{EL1} = \text{find}\,(T_1, \delta_2)$. Subtract the error vector E_{EL1} from the obtained residue digit to correct the error. Go to step 10.

Step 6. If $\delta_1 \neq 0$, $\delta_2 = 0$ and $\delta_3 \neq 0$, then the erroneous digit(s) of the residue are in the EL2. $E_{EL2} = \text{find}\,(T_2, \delta_1)$. Subtract the error vector E_{EL2} from the obtained residue digit to correct the error. Go to step 10.

Step 7. If $\delta_1 = 0$, $\delta_2 \neq 0$ and $\delta_3 \neq 0$, then the erroneous digit(s) of the residue are in the EL3. $E_{EL3} = \text{find}\,(T_3, \delta_2, or \delta_3)$. Subtract the error vector E_{EL3} from the obtained residue digit to correct the error. Go to step 10.

Step 8. If $\delta_1 \neq 0$, $\delta_2 \neq 0$ and $\delta_3 \neq 0$, the erroneous residue digits can be in EL4, EL5, EL6 or EL7. The exact category of the error location is defined as follows:

Step 8.1 If $E_{EL4} = \text{find}\,(T_4, \delta_1) = \text{find}\,(T_1, \delta_2) \cup \text{find}\,(T_2, \delta_3) \neq 0$, then the error is in EL4. Set $E = E_{EL4}$.

Step 8.2 If $E_{EL5} = \text{find}\,(T_5, \delta_2) = \text{find}\,(T_1, \delta_1) \cup \text{find}\,(T_3, \delta_3) \neq 0$, then the error is in EL5. Set $E = E_{EL5}$.

Step 8.3 If $E_{EL6} = \text{find}\,(T_6, \delta_3) = \text{find}\,(T_2, \delta_1) \cup \text{find}\,(T_3, \delta_2) \neq 0$, then the error is in EL6. Set $E = E_{EL6}$.

Step 8.4 If we find $\text{find}\,(T_4, \delta_1) \cap \text{find}\,(T_5, \delta_2) \neq 0$ or $\text{find}\,(T_4, \delta_1) \cap \text{find}\,(T_6, \delta_3) \neq 0$, then the error is in EL7. Set $E = E_{EL7} = \text{find}\,(T_4, \delta_1) \cup \text{find}\,(T_5, \delta_2)$

Step 9. If no or more error location categories are found, then there are more than t residue digits errors. Go to step 10. Otherwise, subtract the error vector E from the resulting residue digit(s) to correct the erroneous reside digit.

Step 10. Complete the procedure.

In step 8, each look-up table is accessed by no more than three different syndromes. The same syndrome can be accessed simultaneously by different look-up tables, but different syndromes cannot access the same look-up table at the same time. Thus, the search for the error vector can be completed in no more than three search cycles. As the search key in the search for these tables are three syndromes P_U size and P_V, size and speed of access to the look-up table are dependent on the number and size of RRNS moduli. This problem can be solved using multi-level look-up table with a smaller width of the input address and less complex addresses decoding schemes. However, the algorithm for calculating the syndrome has a quadratic computational complexity of the number of digits in the working range. To reduce the subtractive complexity of calculating syndromes, we propose to use an approximate method.

4. An approximate method for determining the positional characteristic of a number

To calculate the residue of the division in the method described in paragraph 3, we use an approximate method that allows implementing this operation absolutely correctly.

The essence of the approximate method is based on the use of the relative value of the analyzed numbers to the full range defined by the Chinese remainder theorem, which connects the positional number X with its representation in the residues $(x_1, x_2, \ldots, x_n)$, where X_i is the

smallest non-negative residues of a number in relation to to the moduli of the residue number system $p_1, p_2, \ldots, p_n$ by the following expression

$$X = \left| \sum_{i=1}^{n} \frac{P}{p_i} \left| P_i^{-1} \right|_{p_i} x_i \right|_P \tag{43}$$

where $P = \prod_{i=1}^{n} p_i$, p_i are the RNS moduli, $\left| P_i^{-1} \right|_{p_i}$ is a multiplicative inversion with respect to p_i, and $P_i = \frac{P}{p_i} = p_1 p_2 \cdots p_n$.

If the left and right sides of the expression (43) are divided by the constant P, corresponding to the range of numbers, then we get an approximate value

$$\frac{X}{P} = \left| \sum_{i=1}^{n} \frac{\left| P_i^{-1} \right|_{p_i}}{p_i} x_i \right|_1 \approx \left| \sum_{k_i}^{x_i} \right|_1 \tag{44}$$

where $k_i = \frac{\left| P_i^{-1} \right|_{p_i}}{p_i}$ are the constants of the selected system, and x_i are the digits of the number presented in the RNS, while the value of each sum is in the range $[0, 1)$. The final result of the sum is determined after summing and discarding the integer part of the number, keeping the fractional part of the sum. The fractional part can also be written as $X \bmod 1$, because $X = \lfloor X \rfloor + X \bmod 1$. The number of digits of the fractional part of a number is determined by the maximum possible difference between adjacent numbers. If it is necessary to perform an exact comparison, we need to calculate the value (44), which is the equivalent of converting from RNS to weighted number system. To solve the tasks, it is enough to know approximately the value of the used number X in relation to the dynamic range P, which is performed quite simply, but at the same time the ratio $X = P$, $X < P$ or $X > P$ is correctly determined.

The use of the approximate method makes it possible to replace the computationally complex operation of finding the residue of the division by the range of the system by taking the least significant bits of the number, which makes it possible to reduce the computational complexity from quadratic to linear-logarithmic complexity.

5. Conclusion

The paper presents an efficient modification of the algorithm for detecting and correcting multiple residue errors with a fixed number of calculations of the syndrome, regardless of the size of the set of moduli, using an approximate method. The proposed modification allows reducing the computational complexity of the algorithm for calculating the syndrome from quadratic to linear-logarithmic.

The criteria for uniqueness can be met by selecting moduli from a set of primes to satisfy the desired error correction capability. An extended version of this algorithm with an increase in the number of syndromes depending on the number of information moduli was also proposed to remove the limitation imposed on the size of redundant moduli.

Identifying the location of the error and finding the error vector requires only look-up tables and does not require arithmetic operations. In order to minimize the excess space, an extended algorithm is also proposed in which the number of syndromes and look-up tables increases with the number of information units, but the locations of errors can still be identified without requiring iterative computations.

Acknowledgements The reported study was funded by RFBR, Sirius University of Science and Technology, JSC Russian Railways and Educational Fund "Talent and success", project number 20-37-51004, and Russian Federation President Grant MK-24.2020.9, and SP-3149.2019.5

References

[1] Yang L L and Hanzo L 2002 *IEEE transactions on vehicular technology* **51** 1534–1546

[2] How H, Liew T H, Kuan E L, Yang L L and Hanzo L 2006 *IEEE transactions on vehicular technology* **55** 387–396

[3] Etzel M and Jenkins W 1980 *IEEE Transactions on Acoustics, Speech, and Signal Processing* **28** 538–545

[4] Keller T, Liew T H and Hanzo L 2000 *IEEE Journal on Selected Areas in Communications* **18** 2292–2301

[5] Hanzo L, Liew T H and Yeap B L 2002 *Turbo coding, turbo equalisation, and space-time coding* (Wiley Online Library)

[6] Liew T H, Yang L L and Hanzo L 2006 *IEEE transactions on communications* **54** 1006–1016

[7] Zhang S, Yang L L and Zhang Y 2012 *IEEE transactions on vehicular technology* **61** 1234–1250

[8] Sengupta A and Natarajan B 2013 *IEEE transactions on wireless communications* **12** 2458–2469

[9] Yatskiv V, Yatskiv N, Jun S, Sachenko A and Zhengbing H 2013 The use of modified correction code based on residue number system in wsn *2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)* vol 1 (IEEE) pp 513–516

[10] Beckmann P E and Musicus B R 1993 *IEEE transactions on Signal Processing* **41** 2300–2313

[11] Haron N Z and Hamdioui S 2011 *ACM journal on emerging technologies in computing systems (JETC)* **7** 1–19

[12] Yin P and Li L 2013 A new algorithm for single error correction in rrns *2013 International Conference on Communications, Circuits and Systems (ICCCAS)* vol 2 (IEEE) pp 178–181

[13] Goh V T, Tinauli M and Siddiqi M U 2004 A novel error correction scheme based on the chinese remainder theorem *The Ninth International Conference onCommunications Systems, 2004. ICCS 2004.* (IEEE) pp 461–465

[14] Tang Y, Boutillon E, Jégo C and Jézéquel M 2010 A new single-error correction scheme based on self-diagnosis residue number arithmetic *2010 Conference on Design and Architectures for Signal and Image Processing (DASIP)* (IEEE) pp 27–33

[15] Tay T F and Chang C H 2014 A new algorithm for single residue digit error correction in redundant residue number system *2014 IEEE International Symposium on Circuits and Systems (ISCAS)* (IEEE) pp 1748–1751

[16] Yau S S and Liu Y C 1973 *IEEE Transactions on Computers* **100** 5–11

[17] Krishna H, Lin K Y and Sun J D 1992 *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* **39** 8–17

[18] Mandelbaum D M 1976 *IEEE Transactions on Information Theory*

[19] Goldreich O, Ron D and Sudan M 1999 Chinese remaindering with errors *Proceedings of the thirty-first annual ACM symposium on Theory of computing* pp 225–234

[20] Goh V T and Siddiqi M U 2008 *IEEE Transactions on Communications* **56** 325–330

[21] Tay T F and Chang C H 2015 *IEEE transactions on computers* **65** 396–408

[22] Barsi F and Maestrini P 1973 *IEEE Transactions on Computers* **100** 307–315

[23] Jenkins W K and Altman E J 1988 *IEEE transactions on circuits and systems* **35** 159–167