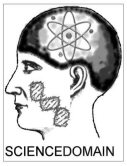
British Journal of Mathematics & Computer Science
11(5): 1-18, 2015, Article no.BJMCS.19903
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org



Computational Algorithms for Syndrome Based Single Error Correction in Residue Number Systems

Hari Krishna Garg[1*]

[1]*Department of Electrical and Computer Engineering, National University of Singapore, Singapore.*

Article Information

DOI: 10.9734/BJMCS/2015/19903
Editor(s):
(1) Doina Bein, Applied Research Laboratory, The Pennsylvania State University, USA.
Reviewers:
(1) Anonymous, R. O. C. Military Academy, Taiwan.
(2) Dominik Strzalka, Rzeszów University of Technology, Poland.
Complete Peer review History: http://sciencedomain.org/review-history/11333

Original Research Article

Received: 12 July 2015
Accepted: 20 August 2015
Published: 09 September 2015

Abstract

Error control via residue number systems continues to attract researchers' attention as evidenced by recent publications dealing with their applications in digital communications and computing. In this paper, we present syndrome based decoding algorithms and analyze their algebraic structure for single error correction in such systems. The mathematical framework is also extended to single error correction and simultaneous multiple error detection. We also bring residue number system product codes under the same framework. Specifically, all the algorithms are based on the computation of a single syndrome value. Computational aspects are also studied along with conditions for the validity of the syndrome based algorithmic approach being described here. Numerous examples are given to illustrate the structure, properties, and decoding procedures associated with the algorithms.

Keywords: Computer arithmetic; Chinese remainder theorem (CRT); residue number systems (RNS); error control; redundant residue number systems (RRNS); single error correction (SEC); multiple error detection (MED).

Abbreviations

CRT : Chinese remainder theorem,
RNS : Residue number systems,
RRNS : Redundant residue number systems,
RNS-PC : Residue Number System-Product Code,
SBEC : Syndrome based error control,
SEC : Single error correction,

**Corresponding author: Email: eleghk@nus.edu.sg;*

MED : *Multiple error detection,*
BE : *Base extension,*
MRC : *Mixed radix conversion,*
MDS : *Maximum Distance Separable,*
SA : *Superfast Algorithm.*

1 Introduction

One of the most popular ways to perform error control in a system, be it a communication or a computing system, is to use an appropriate amount of redundancy. Thus, if **q** is the original information, it is encoded into a codeword **x** such that certain most commonly occurring errors in **c** can be corrected. Most phenomenon that introduce errors are modelled as additive in nature, that is

$$\mathbf{y} = \mathbf{x} + \mathbf{e}. \tag{1}$$

Given **y**, the first step in many syndrome-based error control (**SBEC**) algorithms is to compute syndrome that depends only on the error **e**,

$$\mathbf{s} = \mathbf{S}(\mathbf{y}) = \mathbf{S}(\mathbf{x} + \mathbf{e}) = \mathbf{S}(\mathbf{x}) + \mathbf{S}(\mathbf{e}) = \mathbf{S}(\mathbf{e}). \tag{2}$$

The second step, is to estimate **e** from the syndrome **s**. This can be done via a computational algorithm or a table look-up method. Thus

$$\hat{\mathbf{e}} = \mathbf{T}(\mathbf{s}). \tag{3}$$

Finally, we assume,

$$\mathbf{y} = \hat{\mathbf{x}} + \hat{\mathbf{e}} \tag{4}$$

to decode **y** to

$$\hat{\mathbf{x}} = \mathbf{y} - \hat{\mathbf{e}}. \tag{5}$$

Equivalence between a computational algorithm and table look-up method to carry out the task in (3) is well known in classical coding theory. It is possible to use the method described above to compute more than one possible candidate for $\hat{\mathbf{e}}$ in (3) and hence for $\hat{\mathbf{x}}$ in (5). Such $\hat{\mathbf{x}}$ candidates can then be tested for their validity as a codeword. In our work, we restrict our attention to a single pass where the error $\hat{\mathbf{e}}$ as computed in (3) is unique. Also other methods for decoding, such as 'error trapping,' will also not be studied here.

The above described methodology has also been applied to error control in Chinese remainder theorem (CRT) based residue number systems (RNS) [1]. With redundancy built in for error control, such RNS are termed redundant RNS or RRNS. RRNS continue to attract researchers' attention as evidenced by recent publications dealing with their applications to digital communication and computing systems. The reader is referred to [2-5] for such applications. In addition to RRNS, RNS product codes (RNS-PC) also present an attractive alternative to RRNS for error control in RNS. Both RRNS and RNS-PC will be pursued in this work. When applied to RRNS and RNS-PC, SBEC methods have an additional feature, termed residual effect, to be taken into consideration. The syndrome s in (2) may not depend exclusively on **e**, but also on **x**. As we will establish in this paper, the impact of residual effect depends on algebraic structure of the method used for error control.

In this work, we derive new computational algorithms for single error correction (**SEC**) and its extensions to simultaneous multiple error detection (**MED**) in RRNS and RNS-PC. A rigorous analysis of the residual effect is also carried out. Computational aspects are also studied along with conditions for the validity of the

syndrome based algorithmic approach adopted. Numerous examples are given to illustrate the structure of these algorithms. The key idea of the paper is to use a single syndrome value to carry out the computations required for the task at hand. The approach to use a single syndrome was first described in a recent paper [6] that the current author finds very interesting.

The contributions of this work are two folds. First, many of the existing algorithms for error control in RRNS that utilize multiple syndromes are now reformulated such that they utilize a single syndrome value in a computationally equivalent manner. Collectively, such algorithms for SEC and their extensions for simultaneous MED are called SEC. Second, a new syndrome is defined that can be computed using a superfast algorithm (**SA**) to further simplify the computations involved in error control in RRNS and RNS-PC. This second class of SA for SEC using a single syndrome value is termed **SEC-SA**. The focus in this work is on SEC and SEC-SA for RRNS and RNS-PC.

The organization of this paper is as follows. In section II, the basic framework for RRNS and RNS-PC for error control is presented. Syndrome is defined as a single integer value to be used for determining all single error events. Section III describes residual effect and decoding algorithms for SEC and SEC-MED RRNS. A superfast approach for the computation of syndrome and the corresponding SEC-MED algorithm for RRNS is described in section IV. Conclusions are presented in section V.

It is worthwhile to mention here that we are driven by computational complexity considerations all throughout this work. However we refrain from making statements implying that the computational complexity is the only measure of the overall performance of an algorithm. Further, the need to process large valued integers is avoided in the design of error control algorithms. Instances of such methods that use large valued integers include modulus projection method among many others. Readers may find the research results reported in [7-10] interesting and relevant.

2 Error Control in RRNS & RNS-PC

RNS. A RNS is a finite integer ring $Z(M_K)$ defined by k relatively co-prime moduli $m_1, m_2, ..., m_k$, arranged in ascending order without any loss of generality. The range of the RNS is given by $[0, M_K)$ where

$$M_K = \prod_{i=1}^{k} m_i \ . \tag{6}$$

An integer $X \in [0, M_K)$ in the given RNS is represented as a length k vector $\mathbf{x}$ via the modulo computations,

$$X \leftrightarrow \mathbf{x} = (x_1\ x_2\ ...\ x_k), \tag{7}$$

where

$$x_i \equiv X \text{ modulo } m_i,\ i = 1, 2, ..., k. \tag{8}$$

We assume that all residues are positive. Thus, $0 \le x_i < m_i,\ i = 1, 2, ..., k$. Modulo will be written as 'mod' from here on. Negative integers are converted to equivalent positive values. Thus, $X \equiv -a \bmod m = m - a$, $0 < a < m$. Two observations associated with RNS that play a key role in the decoding algorithms are as follows.

Observation 1. If all residues are equal, say $X \leftrightarrow \mathbf{x} = (a\ a\ ...\ a)$, $0 \le a < m_1$, then $X = a$.

Observation 2. If all residues are equal, say $X \leftrightarrow \mathbf{x} = (-a\ -a\ ...\ -a)$, $0 < a < m_1$, then $X = -a$.

Chinese remainder theorem (CRT). Given **x**, computation of X, $0 \leq X < M_K$, is done via the CRT reconstruction, stated as

$$X \equiv \sum_{i=1}^{k} x_i \cdot t_i \cdot \left(\frac{M_K}{m_i} \right) \mod M_K , \tag{9}$$

where t_i is computed a-priori by solving the congruence

$$t_i \cdot \left(\frac{M_K}{m_i} \right) \equiv 1 \mod m_i , \; i = 1, 2, \ldots, k. \tag{10}$$

It is clear from (10) that

$$\gcd(t_i, m_i) = 1, \; i = 1, 2, \ldots, k. \tag{11}$$

The CRT computation in (9) can be performed in two steps:

Step 1: Compute the permuted residues

$$x_i' \equiv x_i \cdot t_i \mod m_i , \; i = 1, 2, \ldots, k. \tag{12}$$

Step 2: Compute X as

$$X \equiv \sum_{i=1}^{k} x_i' \cdot \left(\frac{M_K}{m_i} \right) \mod M_K . \tag{13}$$

Computation of X from **x** involves large integers when the dynamic range of RNS is large. Another way to express CRT reconstruction of X in (9) is as follows:

$$X \equiv \sum_{i=1}^{k} x_i \cdot t_i \cdot \left(\frac{M_K}{m_i} \right) - \theta_x \cdot M_K . \tag{14}$$

The 'mod M_K' computation in (9) is performed in (14) as a subtraction by a multiple of M_K such that $0 \leq X < M_K$. We note that θ_x is unknown and needs to be determined from the residues **x**.

RRNS. RRNS is obtained by appending $(n - k)$ additional relatively co-prime moduli, $m_{k+1}, \ldots, m_n$, to the RNS defined by moduli $(m_1 \; m_2 \; \ldots \; m_k)$. We further assume that the moduli $(m_1 \; m_2 \; \ldots \; m_k \; m_{k+1} \; \ldots \; m_n)$ are arranged in an ascending order. The RRNS is a (n, k) code. The redundancy of the RRNS is given by

$$M_R = \prod_{i=k+1}^{n} m_i . \tag{15}$$

An integer $X \in [0, M_K)$ in the given RNS is represented as a length n codeword **x** via the modulo computations,

$$X \leftrightarrow \mathbf{x} = (x_1 \; x_2 \; \ldots \; x_k \; x_{k+1} \; \ldots \; x_n), \tag{16}$$

where

$$x_i \equiv X \text{ modulo } m_i,\ i = 1, 2, ..., n. \tag{17}$$

The k residues ($x_1\ x_2\ ...\ x_k$) constitute the information part and ($n - k$) residues ($x_{k+1}\ ...\ x_n$) the parity/redundant part. Let

$$M_N = M_K \cdot M_R. \tag{18}$$

The RRNS as defined here is a maximum distance separable (**MDS**) code having minimum distance

$$d = n - k + 1. \tag{19}$$

Minimum distance is the smallest Hamming distance between two distinct codewords. It also turns out to be the Hamming weight of a non-zero codeword with the smallest Hamming weight [1]. In general, a RRNS that can simultaneously correct α and detect β ($\beta > \alpha$) residue errors has $d = \alpha + \beta + 1$. A SEC RRNS is obtained by setting $d = 3$, $\alpha = 1$. For SEC-MED (multiple error detecting) RRNS, $\alpha = 1$, $\beta = d - 2$. For example, a minimum distance 4 RRNS can simultaneously correct 1 and detect 2 residues in error. .

Example 1. Consider a (4, 2) RRNS defined by ($m_1\ m_2\ m_3\ m_4$) = (11 13 14 15). The legitimate range is [0, 143), M_R = 210. This is a SEC. Given the integer $X = 25 \in [0, 143)$, we have **x** = (3 12 11 10). If we extend this RRNS with a third redundant moduli, say $m_5 = 17$, we get a (5, 2) SEC-DED (double error detecting) RRNS.

An error e_i ($e_i \neq 0$) in the i-th residue leads to the received residue vector **y**,

$$\mathbf{y} = \mathbf{x} + \mathbf{e}, \tag{20}$$

where,

$$y_i \equiv (x_i + e_i) \bmod m_i,\ i = 1, 2, ..., n. \tag{21}$$

For an error, both its location, i such that $e_i \neq 0$, and value, $e_i \neq 0$, are unknown.

Syndrome. Given the residues, **y** = ($y_1\ y_2\ ...\ y_k\ y_{k+1}\ ...\ y_n$), let $\tilde{Y} \in [0, M_K)$ and $Y_R \in [0, M_R)$ represent two integers for the information residues ($y_1\ y_2\ ...\ y_k$) and the parity residues ($y_{k+1}\ ...\ y_n$), respectively. They may be computed using CRT reconstruction or using some other method such as mixed radix conversion (**MRC**). For RRNS the syndrome is an integer δ with value in the range $[0, M_R)$. It is defined as,

$$\delta \equiv (\tilde{Y} - Y_R) \bmod M_R \equiv (\tilde{Y} \bmod M_R - Y_R) \bmod M_R. \tag{22}$$

Given the residue vector **y**, the first step in decoding algorithms for RRNS is syndrome computation in (22).

RNS-PC. For notational consistency, RNS-PC is defined by n moduli ($m_1\ m_2\ ...\ m_n$) such that all codewords when converted to equivalent integer form are divisible by a code-generator integer A, where (A, m_j) = 1, j = 1, 2, ..., n.

$$M_N = \prod_{i=1}^{n} m_i\ . \tag{23}$$

Thus, all integers X in the legitimate range $[0, \lfloor M_N / A \rfloor)$ are first multiplied by A and then converted to a codeword **x** such that

$$A \cdot X \leftrightarrow \mathbf{x} = (x_1\ x_2\ ...\ x_n), \tag{24}$$

where

$$x_i \equiv X \text{ modulo } m_i,\ i = 1, 2, ..., n. \tag{25}$$

Here, $\lfloor M_N / A \rfloor$ is the well-known floor function.

Syndrome. Given the received residues, $\mathbf{y} = (y_1\ y_2\ ...\ y_n)$, let $\tilde{Y} \in [0, M_N)$ represent the integer for the residue vector $\mathbf{y}$. For RNS-PC the syndrome is an integer δ with value in the range $[0, A)$. It is defined as,

$$\delta \equiv \tilde{Y} \mod A. \tag{26}$$

Given the residues, the first step in decoding algorithms for RNS-PC is syndrome computation in (26).

It is clear from the above description that if there are no errors, then $\mathbf{x} = \mathbf{y}$ for either of RRNS or RNS-PC and $\delta = 0$ in (22) for RRNS and in (26) for RNS-PC. Further, though the use of a single syndrome for error control is rather new in RRNS [6], the error control techniques for RNS-PC have always used a single syndrome. Single syndrome based approach to error control for RNS provides a unified framework for algorithm design for seemingly different methodologies. We assume existence of a computational technique to compute $\tilde{Y}$ and Y_R as defined and hence δ in (22) and (26). There exists a significant body of literature on this topic. We refer the readers to [11] for a base extension (BE) method using MRC.

3 SEC Algorithms in RRNS

In this section, we deal with SEC RRNS. We begin our analysis with the effect of information residues on the syndrome manifested via a phenomenon termed 'residual effect'.

Residual effect. We observe that the information and parity residues are treated differently in RRNS while all residues are treated in an identical manner in RNS-PC. Based on (21), if there are one or more errors in the information residues, we have

$$\tilde{Y} \equiv (X + E_I) \bmod M_K = X + E_I - a \cdot M_K. \tag{27}$$

E_I has residues $(e_1\ ...\ e_k)$. Here either $a = 0$ if $X + E_I < M_K$ or $a = 1$ if $X + E_I > M_K$. Similarly, if there are one or more errors in the parity residues, we have

$$Y_R \equiv (X + E_P) \bmod M_R = X \bmod M_R + E_P - b \cdot M_R. \tag{28}$$

E_P has residues $(e_{k+1}\ ...\ e_n)$. Here either $b = 0$ if $X \bmod M_R + E_P < M_R$ or $b = 1$ if $X \bmod M_R + E_P > M_R$. Substituting $\tilde{Y}$ in (27) and Y_R in (28) into (22) we get

$$\begin{aligned} \delta &\equiv \left[\left(X + E_I - a \cdot M_K\right) \bmod M_R - \left(X \bmod M_R + E_P - b \cdot M_R\right)\right] \bmod M_R \\ &= \left(E_I - E_P - a \cdot M_K\right) \bmod M_R . \end{aligned} \tag{29}$$

It is seen from (29) that for the syndrome as calculated in (22) for RRNS, **residual effect** of X is present in the syndrome via the unknown value 'a'. This value gives rise to its own constraints. We note that RNS-PC has the residual effect only due to (27). Thus,

$$\delta \equiv \left(A \cdot X + E - a \cdot M_N\right) \bmod A \equiv (E - a \cdot M_N) \bmod A \tag{30}$$

for RNS-PC. Again, **residual effect** of X is present via the unknown value of 'a'. In the remainder of this section, we describe SEC algorithms for RRNS.

3.1 Minimum Distance 3 RRNS

In this sub-section, we deal with minimum distance 3 RRNS with $n = k + 2$. The two known SEC algorithms that we compare in this work are available in [12] and [6]. We show them to be computationally equivalent. Hence it is reasoned that the conditions for their validity are also identical. Further a computational algorithm is described for SEC based on a single syndrome.

Algorithm 1 [12]

Given the residues, $\mathbf{y} = (y_1\ y_2\ ...\ y_k\ y_{k+1}\ y_{k+2})$, let $\tilde{Y} \in [0, M_K)$ represent the integer for the information residues $(y_1\ y_2\ ...\ y_k)$. For the SEC decoding algorithm of [12], step 1 is syndrome computation. It computes two values (eqn 7.3.1 [1]),

$$\delta_1 \equiv (\tilde{Y} - y_{k+1}) \bmod m_{k+1} \equiv (\tilde{Y} \bmod m_{k+1} - y_{k+1}) \bmod m_{k+1} \tag{31}$$

$$\delta_2 \equiv (\tilde{Y} - y_{k+2}) \bmod m_{k+2} \equiv (\tilde{Y} \bmod m_{k+2} - y_{k+2}) \bmod m_{k+2}. \tag{32}$$

In step 2, error correction is performed as a computational procedure. If $\delta_1 = \delta_2 = 0$, $\mathbf{y}$ is error-free. If only one of $(\delta_1\ \delta_2)$ is 0, the corresponding parity digit is assumed to be in error and corrected. If both the syndromes are non-zero, then an error is assumed in i-th information residue and value of error residue is computed if the assumption is found to be valid, $i = 1, 2, ..., k$.

Algorithm 2 [6]

The step 1 of this algorithm computes the syndrome as defined in (22). It is clear that (29) also holds. Thus, for the syndrome in (22) or (29), residual effect of X is present via the unknown value of 'a' in (29). In step 2, error correction is performed via a table look-up.

Algorithm 1 avoids processing large valued integers by using BE and MRC techniques for computing δ_1 and δ_2 in (31) and (32), while Algorithm 2 uses CRT to first compute $\tilde{Y}$ and Y_R before carrying out mod M_R operation in (22). We emphasize that the above described algorithms compute essentially the same quantities though they follow different computational approaches. In that regard, they are equivalent. We observe that the phrases 'computational equivalence' or 'algorithm equivalence' are known in Computer Science [13].

Theorem 1. The first steps of algorithms 1 and 2 are equivalent.

Proof. With M_R as defined in (15), CRT establishes equivalence between residues δ_1 and δ_2 expressed modulo m_{k+1} and m_{k+2} in (31) and (32), respectively, and the residue δ modulo M_R in (22). It is seen that, $\delta_1 \equiv \delta \bmod m_{k+1}$ and $\delta_2 \equiv \delta \bmod m_{k+2}$.

Example 2. For the example in [6], the (5, 3) RRNS is given by moduli (7 9 11 13 17), $M_R = 221$. Let $\mathbf{y}$ = (5 5 9 8 8). As per the computation in [6], $\tilde{Y} = 383$, $Y_R = 8$ and $\delta \equiv (\tilde{Y} - Y_R) \bmod M_R = (383 - 8) \bmod 221 = 154$. As per Algorithm 1, $\delta_1 \equiv (\tilde{Y} \bmod m_{k+1} - y_{k+1}) \bmod m_{k+1} \equiv (6 - 8) \bmod 13 = 11$, $\delta_2 \equiv (\tilde{Y} \bmod m_{k+2} - y_{k+2}) \bmod m_{k+2} \equiv (9 - 8) \bmod 17 = 1$. The equivalence is seen as $\delta_1 = 11 \equiv \delta \bmod 13 = 154 \bmod 13 = 11$ and $\delta_2 = 1 \equiv \delta \bmod 17 = 154 \bmod 17 = 1$.

Given the equivalence of step 1 of two algorithms, we now examine conditions for the validity of their step 2. Step 2 of both algorithms provides the same output, realized via computations in [12] and table look-up in [6]. Since the inputs to the step 2 of both algorithms are equivalent, it is imperative that the conditions for

effectively countering residual effect in one algorithm must apply to the other. Therefore in addition to the redundancy constraint $M_R = m_{k+1} \cdot m_{k+2}$, the necessary and sufficient conditions for algorithms 1 and 2 to provide a correct solution can be stated as follows.

Theorem 2. If the information moduli m_i and m_j are such that there do not exist integers n_i and n_j, $0 < n_i < m_i$, $0 < n_j < m_j$ that satisfy

$$n_i \cdot m_j + n_j \cdot m_i = m_{k+1} \cdot m_{k+2},\ i, j = 1, 2, \ldots, k; \tag{33}$$

then solutions in step 2 of algorithms 1 and 2 are correct under the assumption that at most one error has occurred.

Note that this is same as Theorem 8.1 in [1, p 191] stated for Algorithm 1. The condition

$$m_{k+1} \cdot m_{k+2} > 2 \cdot m_i \cdot m_j - m_i - m_j \tag{34}$$

is a sufficient condition to (33) obtained from (33) by replacing integers n_i and n_j by their maximum values $m_i - 1$ and $m_j - 1$, respectively. It may also be used to eliminate those pairs of information moduli m_i and m_j that need not be tested for (33). In our work, we have found many cases of RRNS that satisfy (33) as well as those that don't. Some of them will be mentioned in the following. We note that the RRNS in example 2 (taken from [6]) satisfies the sufficient condition in (34). Further, contrary to what is stated in [6], we reassert that for the step 2 of Algorithm 2 to be valid, conditions in (33) or (34) must be satisfied. There is an oversight in the analysis of case 1 of theorem 2 in [6]. The correct expression that must have been used is $E_i = a_i M_K / m_i$ instead of $E_i = a_i M_K M_R / m_i$.

Example 3. Consider the (4, 2) RRNS in example 1. Here, $m_1 \cdot m_2 = 143 < m_3 \cdot m_4 = 210 < 2 \cdot m_1 \cdot m_2 - m_1 - m_2 = 262$. Further, $12 \cdot m_1 + 6 \cdot m_2 = 132 + 78 = m_3 \cdot m_4 = 210$. Thus the condition for a unique association between the syndrome and the error (location and value) is violated. For the codeword $X = 136 \leftrightarrow \mathbf{x} = (4\ 6\ 10\ 1)$, let a single error be (0 11 0 0). Thus, $\mathbf{y} = (4\ 4\ 10\ 1)$. The corresponding syndrome is

$$\delta \equiv (\tilde{Y} \mod M_R - Y_R) \mod M_R = (4 \mod 210 - 136) \mod 210 = 78.$$

Similarly, for the codeword $X = 16 \leftrightarrow \mathbf{x} = (5\ 3\ 2\ 1)$, let a single error be (1 0 0 0). Thus, $\mathbf{y} = (6\ 3\ 2\ 1)$. The corresponding syndrome is

$$\delta \equiv (\tilde{Y} \mod M_R - Y_R) \mod M_R = (94 \mod 210 - 16) \mod 210 = 78.$$

It is not surprising that syndrome $\delta = 78$ is obtained for multiple codewords when they are corrupted by single error events (1 0 0 0) or (0 11 0 0). Similarly, syndrome $\delta = 132$ is obtained for multiple codewords when they are corrupted by either the single error event (10 0 0 0) or (0 2 0 0). In all such cases, there exists no unique relationship between the syndrome δ and single error events. Hence neither algorithm 1 nor algorithm 2 can be used for error correction with these moduli. As per [6], Algorithm 2 can be used for SEC in this case. This is erroneous.

Example 4. Consider the (8, 6) RRNS defined by the residues (23 25 27 29 31 32 37 43). Here, $m_7 \cdot m_8 = 1591$. We need to test the condition of Theorem 2 for the residue pairs $(m_6\ m_5) = (32\ 31)$, $(m_6\ m_4) = (32\ 29)$, $(m_6\ m_3) = (32\ 27)$, $(m_5\ m_4) = (31\ 29)$, and $(m_5\ m_3) = (31\ 27)$ as for each of them $2 \cdot m_i \cdot m_j - m_i - m_j > 1591$. These values are (1921 1795 1669 1738 1616). A quick evaluation of the linear Diophantine equation in (33) leads to no solution in any of the 5 cases. The obtained solutions are $10 \cdot m_6 + 9 \cdot m_5 = 599$, $18 \cdot m_6 + 3 \cdot m_4 = 663$, $5 \cdot m_6 + 21 \cdot m_3 = 727$, $25 \cdot m_5 + 26 \cdot m_4 = 1529$, and $13 \cdot m_5 + 13 \cdot m_3 = 754$. Hence this RRNS can be decoded using either of the two algorithms.

Example 5. Consider the (8, 6) RRNS defined by the residues (23 25 27 29 31 32 41 43). Here, $m_7 \cdot m_8 = 1763$. We need to test the condition (33) of Theorem 2 for the residue pairs $(m_6\, m_5) = (32\ 31)$ and $(m_6\ m_4) = (32\ 29)$ as for each pair $2 \cdot m_i \cdot m_j - m_i - m_j > 1763$. These values are 1921 and 1795. A quick evaluation of the linear Diophantine equation in (33) leads to a solution in both cases given by $27 \cdot m_6 + 29 \cdot m_5 = 864 + 899 = 1763$ and $27 \cdot m_6 + 31 \cdot m_4 = 864 + 899 = 1763$. Hence, neither of the two algorithms can be used for decoding this RRNS. There will be several single error events each with syndrome values of 864 and 899.

Algorithm 1 uses two syndromes in step 1 and a computational procedure for determining error location/value in step 2. We now describe a computational procedure for step 2 of Algorithm 1 that uses a single syndrome as in (29).

Step 2 of Algorithm 1 (single syndrome based): Error computation

Input: Syndrome integer δ.

Output: Corrected residue once error (i and e_i such that $e_i \neq 0$) is known.

A. No Error. If $\delta = 0$, no error is declared. **STOP**.

B. Error in parity residue. Compute

$$\delta_1 \equiv \delta \bmod m_{k+1},$$
$$\delta_2 \equiv \delta \bmod m_{k+2}.$$

If only one of δ_1 and $\delta_2 = 0$, then parity residue with non-zero δ_1 or δ_2 is in error. If $\delta_1 \neq 0$ then $e_{k+1} = -\delta_1$. If $\delta_2 \neq 0$ then $e_{k+2} = -\delta_2$. Go to **D**.

C. Error in information residue. Compute

$$\Delta \equiv (M_K)^{-1} \cdot \delta \bmod M_R.$$

For $j = 1, 2, ..., k$, compute

$$e_{1,j} \equiv m_j \cdot \Delta \bmod M_R,$$
$$e_{2,j} = M_R - e_{1,j}.$$

If either one of $e_{1,j}$ or $e_{2,j} \in (0, m_i)$ then j-th residue is declared to be in error,

where

$$e_j \equiv (t_j)^{-1} \cdot e_{1,j} \bmod m_j \equiv [(M_K / m_j) \bmod m_j] \cdot e_{1,j} \bmod m_j,$$

if $0 < e_{1,j} < m_j$, and

$$e_j \equiv m_j - (t_j)^{-1} \cdot e_{2,j} \bmod m_j \equiv m_j - [(M_K / m_j) \bmod m_j] \cdot e_{2,j} \bmod m_j,$$

if $0 < e_{2,j} < m_j$. Next, go to **D**.

D. Single residue correction. For i-th residue in error, $i \in (1\ 2\ \ldots\ k + 2)$, error correction is carried out as

$$x_i \equiv (y_i - e_i) \bmod m_i.$$

E. End

Scalars t_i, used in CRT reconstruction of $\tilde{Y}$, are obtained by solving the congruence, $t_i \cdot (M_K/m_i) \equiv 1 \bmod m_i$, $i = 1, 2, \ldots, k$. Thus, $(t_i)^{-1} \equiv (M_K/m_i) \bmod m_i$.

We illustrate this computational step with an example.

Example 6. This is a continuation of example 2. The (5, 3) RRNS is given by moduli (7 9 11 13 17), M_K = 693, M_R = 221. We need the following one-time pre-computations for steps 1 and 2: (t_1 t_2 t_3) = (1 2 7) to compute $\tilde{Y}$ from residues (y_1 y_2 y_3) and (t_4 t_5) = (10 4) to compute Y_R from residues (y_4 y_5). Also, (t_1^{-1} t_2^{-1} t_3^{-1}) = (1 5 8) and $(M_K)^{-1} \bmod M_R$ = 140. Let **y** = (5 4 1 8 8). In this case, $\tilde{Y}$ = 166, Y_R = 8, and $\delta \equiv (\tilde{Y} - Y_R) \bmod M_R = (166 - 8) \bmod 221 = 158$. As $\delta \neq 0$, we proceed as follows:

Step B.

$$\delta_1 \equiv \delta \bmod m_{k+1} = 158 \bmod 13 = 2,$$
$$\delta_2 \equiv \delta \bmod m_{k+2} = 158 \bmod 17 = 5.$$

As both δ_1 and δ_2 are non-zero, we assume single error in an information residue.

Step C. The various computations are

$$\Delta \equiv (M_K)^{-1} \cdot \delta \bmod M_R = 140 \cdot 158 \bmod 221 = 20.$$

$i = 1$

$$e_{1,1} \equiv m_1 \cdot \Delta \bmod M_R = 7 \cdot 20 \bmod 221 = 140,$$
$$e_{2,1} = M_R - e_{1,1} = 221 - 140 = 81.$$

$i = 2$

$$e_{1,2} \equiv m_2 \cdot \Delta \bmod M_R = 9 \cdot 20 \bmod 221 = 180,$$
$$e_{2,2} = M_R - e_{1,2} = 221 - 180 = 41.$$

$i = 3$

$$e_{1,3} \equiv m_3 \cdot \Delta \bmod M_R = 11 \cdot 20 \bmod 221 = 220,$$
$$e_{2,3} = M_R - e_{1,3} = 221 - 220 = 1.$$

Since $0 < e_{2,3} < m_3$, an error is declared in 3rd residue, the error value being

$$e_3 \equiv m_3 - (t_3)^{-1} \cdot e_{2,3} \bmod m_3 \equiv m_3 - [(M_K/m_3) \bmod m_3] \cdot e_{2,3} \bmod m_3$$
$$= 11 - 8 \cdot 1 \bmod 11 = 3.$$

Step D. The error correction in 3rd residue is carried out as

$$x_3 \equiv (y_3 - e_3) \bmod m_3 = (1 - 3) \bmod 11 = 9.$$

This completes our analysis of syndrome based minimum distance 3 SEC algorithms in RRNS.

3.2 Algorithms for SEC-MED in RRNS

For SEC-MED algorithms, we need to determine the location and value of single errors and simultaneously carry out MED. In the following, we describe a SEC-MED algorithm for RRNS. If no errors are present, then $\delta = 0$.

If only a single error occurs in the i-th parity residue, $i = k + 1, ..., n$, then

$$E_P = E_i = [(e_i \cdot s_i) \bmod m_i] \cdot (M_R / m_i), \tag{35}$$

$$E_I = 0,\ a = 0,\ \delta \equiv -E_P \bmod M_R . \tag{36}$$

Integers s_i are CRT reconstruction integers used in computation of an integer from its residues defined by moduli ($m_{k+1} ... m_n$) over $[0, M_R)$. It is clear from (35) that

$$e_i \equiv -\delta \bmod m_i,$$

$$e_j \equiv \delta \bmod m_j = 0,\ j = k + 1, ..., i - 1, i + 1, ... n. \tag{37}$$

An alternative way to express write (37) is

$$E_i \equiv \delta \bmod (M_R / m_i) = 0. \tag{38}$$

If only a single error occurs in the i-th information residue, $i = 1, 2, ..., k$, then

$$E_I = E_i = (e_i \cdot t_i) \bmod m_i \cdot (M_K / m_i) = e_i' \cdot \left(\frac{M_K}{m_i} \right), \tag{39}$$

$$E_P = 0, \tag{40}$$

$$\delta \equiv \left(E_i - a \cdot M_K\right) \bmod M_R = \left(e_i' \cdot \left(\frac{M_K}{m_i} \right) - a \cdot M_K \right) \bmod M_R . \tag{41}$$

Based on (39)-(41), an algorithm for computing the error location and value in information residues is derived as follows. Compute:

$$\Delta \equiv M_K^{-1} \cdot \delta \bmod M_R \equiv \left(e_i' m_i^{-1} - a\right) \bmod M_R . \tag{42}$$

For $j = 1, 2, ..., k$, assume error in j-th residue and compute

$$\Delta_j \equiv m_j \cdot \Delta \bmod M_R \equiv \left(e_i' m_i^{-1} \cdot m_j - a \cdot m_j\right) \bmod M_R . \tag{43}$$

When $j = i$,

$$\Delta_i \equiv \left(e_i' - a \cdot m_i\right) \bmod M_R . \tag{44}$$

Two cases arise due to residual effect, $a = 0$ resulting in

$$e_{i,0}' \equiv \Delta_i \bmod M_R = e_i' \tag{45}$$

or $a = 1$, resulting in $(e_i' - m_i) \equiv \Delta_i \bmod M_R$. Rearranging,

$$e_{i,1}' = m_i - e_i' \equiv -\Delta_i \bmod M_R = M_R - \Delta_i = M_R - e_{i,0}' \tag{46}$$

for $a = 1$. In all other cases when $j \neq i$, (43) holds.

It is clear that one of the two values in (45) or (46) belongs to the range $(0, m_i)$. We further note that if $0 \leq \Delta_i \bmod M_R < m_i$ or $-m_i < \Delta_i \bmod M_R < 0$ then $\Delta_i \bmod m_{k+l} = \Delta_i \bmod M_R$, $l = 1, 2, ..., n - k$. This is seen via observations 1 and 2. Expressions in (44)-(46) become our test for identifying i, location of the error, and e_i, value of the error. We claim that the solutions to (43) for $a = 0$ or $a = 1$ do not satisfy the conditions $0 < e'_{j,0} < m_j$ or $0 < e'_{j,1} < m_j$ for $j \neq i$. This claim can be proven in a straightforward manner. The framework for SEC-MED in RRNS is complete. The algorithm can be described as follows.

Algorithm for SEC-MED in RRNS (single syndrome based)

Input: Received residues $\mathbf{y} = (y_1 \ldots y_n)$.

Output: Corrected residues if up to 1 error occurs; Errors detected if up to $d - 2$ errors occur.

Step 1. Compute syndrome value δ in (22).

Step 2. Error Computation

A. No Error. If $\delta = 0$, then declare "no error occurred." **STOP**.

B1. Error in parity residue: Approach 1. For $j = k + 1, ..., n$, compute

$$\delta_j \equiv \delta \bmod m_j.$$

If exactly one of $\delta_j \neq 0$, then declare "1 error in parity residue." Say, for $j = i$, $\delta_i \neq 0$ then i-th parity residue is in error, $e_i = -\delta_i \bmod m_i$. Next, go to **E**.

B2. Error in parity residue: Approach 2. For $j = k + 1, ..., n$, compute

$$\delta_j \equiv \delta \bmod (M_R / m_j).$$

If $\delta_j = 0$, then declare "1 error in parity residue." Say, for $j = i$, $\delta_i = 0$ then i-th parity residue is in error, $e_i = -\delta \bmod m_i$. Next, go to **E**.

C. Error in information residue. Compute

$$\Delta \equiv M_K^{-1} \cdot \delta \bmod M_R .$$

For $j = 1, 2, ..., k$, compute

$$e_{j,0} \equiv m_j \cdot \Delta \bmod M_R,$$

$$e_{j,1} = M_R - e_{j,0}.$$

If either one of $e_{j,0}$ or $e_{j,1} \in (0, m_j)$ then declare the j-th residue to be in error,

where

$$e_j \equiv (t_j)^{-1} \cdot e_{j,0} \bmod m_j,$$

if $0 < e_{j,0} < m_j$, and

$$e_j \equiv m_j - (t_j)^{-1} \cdot e_{j,1} \bmod m_j,$$

if $0 < e_{j,1} < m_j$. Next, go to **E**.

D. Multiple error detection. Declare "more than 1 error detected." **STOP**.

E. Single residue correction. For i-th residue in error, $i \in (1\ 2\ \ldots\ n)$, error correction is carried out as

$x_i \equiv (y_i - e_i) \bmod m_i$.

F. End

Two equivalent approaches to carry out the computation in step B 'Error in parity residue' are given. Either one may be used.

We note that the above described single syndrome based algorithm for SEC-MED in RRNS is computationally equivalent to the multiple syndrome based algorithm for SEC-MED in [1]. Hence the conditions for the validity of the above algorithm are identical to those in [1]. Though the necessary and sufficient conditions are quite cumbersome, the sufficient condition is not. It is same as the sufficient condition in (34).

Example 7. Consider a (10, 6) RRNS defined by $(m_1\ m_2\ \ldots\ m_{10})$ = (23 25 27 29 31 32 67 71 73 79) with d = 5, α = 1, and β = 3. Clearly $m_7 \cdot m_8$ = 4,757 > 2 · $m_6 \cdot m_5 - m_6 - m_5$ = 1,921, and therefore the sufficient condition in (34) is satisfied. $M_K = \prod_{i=1}^{6} m_i$ = 446,623,200 and $M_R = \prod_{i=7}^{10} m_i$ = 27,433,619. Let X = 400,000,000, then **x** = (8 0 22 13 25 0 17 58 4 11). Assume that one error takes place in the first residue, and the received vector is **y** = (0 0 22 13 25 0 17 58 4 11). Based on the information part (0 0 22 13 25 0) and parity part (17 58 4 11),

Step 1. Syndrome is computed as

$\delta \equiv (\tilde{Y} \bmod M_R - Y_R) \bmod M_R$

= (225,234,400 mod 27,433,619 – 15,929,334) mod 27,433,619

= 17,269,733.

Step 2. Error computation

A. $\delta \neq 0$. At least error has occurred.

B1. Error in parity residue: Approach 1. Compute $\delta_j \equiv \delta \bmod m_j$, j = 7, ..., 10,

$\delta_7 \equiv \delta \bmod m_7$ = 17,269,733 mod 67 = 14

$\delta_8 \equiv \delta \bmod m_8$ = 17,269,733 mod 71 = 48

$\delta_9 \equiv \delta \bmod m_9$ = 17,269,733 mod 73 = 50

$\delta_{10} \equiv \delta \bmod m_{10}$ = 17,269,733 mod 79 = 17

Since more than one of δ_j, j = 7, ..., 10, is non-zero, we go to next step.

B2. Error in parity residue: Approach 2. Compute $\delta_j \equiv \delta \bmod (M_R / m_j)$, j = 7, ..., 10,

$\delta_7 \equiv 17{,}269{,}733 \bmod 409{,}457 = 72{,}539$

$\delta_8 \equiv 17{,}269{,}733 \bmod 386{,}389 = 268{,}617$

$\delta_9 \equiv 17{,}269{,}733 \bmod 375{,}803 = 358{,}598$

$\delta_{10} \equiv 17{,}269{,}733 \bmod 347{,}261 = 253{,}944.$

Since none of $\delta_j = 0$, we go to next step.

C. Error in information residue. Compute $\Delta \equiv M_K^{-1} \cdot \delta \bmod M_R$

$\Delta \equiv 5{,}014{,}626 \cdot 17{,}269{,}733 \bmod 27{,}433{,}619 = 10{,}734{,}894.$

$j = 1$

$e_{1,0} \equiv m_1 \cdot \Delta \bmod M_R = 23 \cdot 10{,}734{,}894 \bmod 27{,}433{,}619 = 27{,}433{,}610.$

$e_{1,1} = M_R - e_{1,0} = 9.$

Since $0 < e_{1,1} < m_1$, an error is declared in 1st information residue with

$e_1 \equiv m_1 - (t_1)^{-1} \cdot e_{1,1} \bmod m_1 = 23 - [(M_K / m_1) \bmod m_1] \cdot e_{1,1} \bmod m_1$
$= 23 - 6 \cdot 9 \bmod 23 = 15.$

Next, go to **E**.

E. Single residue correction. For the 1st residue in error, error correction is performed as

$x_1 \equiv (y_1 - e_1) \bmod m_1 = (0 - 15) \bmod 23 = 8.$

Now assume that two errors take place in the first and third residue digit and the received vector is **y** = (0 0 23 13 25 0 17 58 4 11). Based on the information part (0 0 23 13 25 0) and the parity part (17 58 4 11) we compute the syndrome as

$\delta \equiv (\tilde{Y} \bmod M_R - Y_R) \bmod M_R$

$= (109{,}443{,}200 \bmod 27{,}433{,}619 - 15{,}929{,}334) \bmod 27{,}433{,}619$

$= 11{,}213{,}009.$

Following the decoding algorithm, we check the consistency for j = 1, ..., 6, and find that there is no consistent solution. Therefore, more than one error is detected.

4 Superfast Algorithm for SEC-MED in RRNS

In this section, we deal with SEC-SA, a superfast algorithm for SEC-MED RRNS. This is a generalization of the minimum distance 3 SEC-SA described in [14]. SEC-SA is based on the CRT reconstruction expression in (14). It is restated again as it is extensively utilized in this section.

$$X \equiv \sum_{i=1}^{k} x_i \cdot t_i \cdot \left(\frac{M_K}{m_i} \right) \quad - \theta_x \cdot M_K \,, \; 0 \le \theta_x < k. \tag{47}$$

There is further uncertainty here due to unknown θ_x in (47). To counter it, we introduce an additional redundant moduli m_{n+1} to create an (n + 1, k) RRNS from the original (n, k) RRNS such that

$$m_{n+1} > k. \tag{48}$$

The (n + 1, k) RRNS will still be treated as a minimum distance $d = n - k + 1$ code, where moduli m_{n+1} is used exclusively to manage the uncertainty in (47).

Consider a residue vector $\mathbf{y} = (y_1 \ldots y_k \, y_{k+1} \ldots y_n \, y_{n+1})$ for a codeword $\mathbf{x}$. According to $\mathbf{y}$, we may compute $\tilde{\tilde{Y}}$ as

$$\tilde{\tilde{Y}} \equiv \sum_{i=1}^{k} y_i \cdot t_i \cdot \left(\frac{M_K}{m_i} \right) \quad - \theta_y \cdot M_K \,, \; 0 \le \theta_y < k \tag{49}$$

and proceed as before. The SEC-SA is based on the expression for $\hat{Y}$ defined as follows,

$$\hat{Y} \equiv \sum_{i=1}^{k} y_i \cdot t_i \cdot \left(\frac{M_K}{m_i} \right). \tag{50}$$

It is clear from (49) that

$$\hat{Y} \equiv \tilde{\tilde{Y}} + \theta_y \cdot M_K \,. \tag{51}$$

The first step then in the SEC-SA is to compute a quantity D:

$$D \equiv (\hat{Y} - Y_R) \bmod M_R \equiv (\hat{Y} \bmod M_R - Y_R) \bmod M_R. \tag{52}$$

Here Y_R is the integer obtained by combining the parity residues y_l, $l = k + 1, \ldots, n + 1$. It was used earlier in (22) also. It is possible to avoid large integer arithmetic by computing $\hat{Y} \bmod M_R$ directly rather than computing $\hat{Y}$ first and then taking mod M_R. The modified syndrome to be used in SEC-SA is then defined as

$$\delta \equiv M_K^{-1} \cdot D \bmod M_R \,. \tag{53}$$

Substituting for $\hat{Y}$ from (51), $\tilde{\tilde{Y}}$ from (27) and D from (52) into (53), we get

$$\delta \equiv \left[M_K^{-1} (E_I - E_P) + (\theta_y - a) \right] \bmod M_R \,. \tag{54}$$

As before, three cases follow.

Case 1. No error in $\mathbf{y}$. In this case, $E_I = E_P = a = 0$ and

$$\delta \equiv \theta_y \bmod M_R = \theta_y \,, \; 0 \le \theta_y < k. \tag{55}$$

Case 2. An error in information part in **y**. Let *i*-th information residue be in error. In this case, E_I is same as in (39), $E_P = 0$, and

$$\delta = \left\lfloor e_i' \cdot m_i^{-1} + (\theta_y - a) \right\rfloor \bmod M_R, -1 \leq \theta_y - a < k. \tag{56}$$

Thus,

$$m_i \cdot \delta = \left\lfloor e_i' + m_i \cdot (\theta_y - a) \right\rfloor \bmod M_R, -1 \leq \theta_y - a < k. \tag{57}$$

Finally, (57) leads to $-m_i < m_i \cdot \delta < k \cdot m_i$. Since we process only positive integers in modulo arithmetic, we may write, $0 \leq m_i \cdot \delta < k \cdot m_i$ or $M_R - m_i < m_i \cdot \delta < M_R$.

Case 3. An error in parity part in **y**. In this case, $E_I = a = 0$ and

$$\delta \equiv \left[- M_K^{-1} \cdot E_P + \theta_y \right] \bmod M_R, 0 \leq \theta_y < k. \tag{58}$$

Another way to express (58) for an error in the *i*-th parity residue, $i = k + 1, \ldots, n + 1$, is

$$e_i \equiv \theta_y - M_K^{-1} \cdot \delta \bmod m_i, \tag{59}$$

$$e_j \equiv \delta \bmod m_j = \theta_y, j = k + 1, \ldots, i - 1, i + 1, \ldots n. \tag{60}$$

We may also write (58) as

$$\delta_i \equiv \delta \bmod (M_R / m_i) = \theta_y. \tag{61}$$

Based on the above analysis of error events, a complete SEC-SA for RRNS can now be described as follows:

Algorithm for SEC-SA for SEC-MED in RRNS (single syndrome based)

Input: Received residues $\mathbf{y} = (y_1 \ldots y_n y_{n+1})$.

Output: Corrected residues if up to 1 error occurs; Errors detected if up to $d - 2$ errors occur.

Step 1. Compute the quantity D in (52) and syndrome value δ in (53).

Step 2. Error Computation

A. No Error. If $0 \leq \delta < k$, then declare "no error occurred." **STOP**.

B. Error in parity residue: Approach 1. For $l = k + 1, \ldots, n + 1$, compute

$\delta_l \equiv \delta \bmod (M / m_l)$.

If exactly one of $0 \leq \delta_l < k$, then declare "1 error in parity residue." Say, for $l = i$, $0 \leq \delta_l < k$ then *i*-th parity residue is in error, $\theta_y = \delta_l$ and $e_i \equiv \theta_y - M_K^{-1} \cdot \delta \bmod m_i$. Next, go to **E**.

C. Error in information residue.

For $j = 1, 2, ..., k$, compute
$e_{j,0} \equiv m_j \cdot \delta \bmod M_R$,
$e_{j,1} = M_R - e_{j,0}$.

If either $0 < e_{j,0} < k \cdot m_j$ or $0 < e_{j,1} < m_j$ then declare the j-th residue in error,

Where

$e_j \equiv (t_j)^{-1} \cdot e_{j,0} \bmod m_j$,

if $0 < e_{j,0} < k \cdot m_j$, and

$e_j \equiv m_j - (t_i)^{-1} \cdot e_{j,1} \bmod m_j$,

if $0 < e_{j,1} < m_j$. Next, go to **E**.

D. Multiple error detection. Declare "more than 1 error detected." **STOP**.

E. Single residue correction. For i-th residue in error, $i \in (1\ 2\ \ldots\ n+1)$, error correction is carried out as

$x_i \equiv (y_i - e_i) \bmod m_i$.

F. END.

The above SEC-SA has been described for RRNS. A similar SEC-SA can be also described for RNS-PC. We end this section by stating that the SEC-MED and SEC-SA algorithms described here are computational in nature. It is straightforward to describe an equivalent implementation that is based on a table look-up once the syndrome value is computed in step 1 of the decoding algorithms.

5 Conclusion

In this work, we have described mathematical construction and algorithms associated with syndrome based single error correcting codes for RRNS. All such algorithms are described within the framework of a single syndrome used for error correction as well as simultaneous error detection. Examples are also provided to illustrate the various features and procedures associated with the various algorithms. Finally, a superfast algorithm is described that simplifies computations at the expense of an additional moduli.

Competing Interests

Author has declared that no competing interests exist.

References

[1] Krishna H, Krishna B, Lin KY, Sun JD. Computational number theory and digital signal processing: Fast algorithms and error control techniques. CRC Press, Boca Raton, USA; 1994.

[2] Sengupta A, Natarajan B. Performance of systematic RRNS based space-time block codes with probability-aware adaptive demapping. IEEE Transactions on Wireless Communications. 2013; 12(5):2458-2469.

[3] Zhang S, Zhang Y, Yang LL. Redundant residue number system based multicarrier DS-CDMA for dynamic multiple-access in cognitive radios. IEEE Vehicular Technology Conference, Japan. 2011; 1–5.

[4] Sengupta A, Zhu D, Natarajan B. On the performance of redundant residue number system codes assisted STBC design. International Conference on Computing, Networking and Communications (ICNC) 2012, Wireless Communications Symposium. 2012;1051-1055.

[5] Haron NZ, Hamdioui S. Using RRNS codes for cluster faults tolerance in hybrid memories. 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, USA. 2009;86-93.

[6] Tay TF, Chang CH. A new algorithm for single residue digit error correction in redundant residue number system. IEEE International Symposium on Circuits and Systems (ISCAS), Australia. 2014; 1748-1751.

[7] Mandelbaum DM. On a class of arithmetic codes and a decoding algorithm. IEEE Transactions on Information Theory. 1976;IT-22:85-88.

[8] Soderstrand MA, Jenkins WK, Julien GA, Taylor FJ. Modern applications of residue number system arithmetic to digital signal processing. New York, IEEE Press; 1986.

[9] Shenoy AP, Kumaresan R. Fast base extension using a redundant modulus in RNS. IEEE Transactions on Computers. 1989;C-38:292-297.

[10] Mandelbaum DM. An approach to an arithmetic analog of Berlekamp's algorithm. IEEE Transactions on Information Theory. 1984;IT-30:758-762.

[11] Jenkins WK, Altman EJ. Self-checking properties of residue number error checkers based on mixed radix conversion. IEEE Transactions on Circuits & Systems. 1988;35:159-167.

[12] Krishna H, Lin KY, Sun JD. A coding theory approach to error control in redundant residue number systems. Part I: Theory and Single Error Correction, IEEE Transactions on Circuits & Systems. 1992; 39:8-17.

[13] Blass A, Dershowitz N, Gurevich Y. When are two algorithms the same? arXiv.org; 2008.

[14] Sun JD, Krishna H, Lin KY. A superfast algorithm for single error correction in RRNS and hardware implementation. Journal of VLSI Signal Processing. 1993;6:259-269.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/11333