

MODIFIED MODULUS PROJECTION ALGORITHM FOR ERROR CONTROL IN REDUNDANT RESIDUE NUMBER SYSTEM CODE

*K. A. AMUSA AND O. O. NUGA

Department of Electrical Electronic Engineering,
Federal University of Agriculture, Abeokuta

*Corresponding author: amusaakinwale@gmail.com; Tel: +2347067936448

ABSTRACT

This paper presents a modified modulus projection algorithm for decoding Redundant Residue Number System (RRNS) codes. RRNS code is maximum-minimum distance block codes. RRNS has found wider applications in the field of signal processing especially for error control. The essence of this modification is to reduce the required computation overhead and speed up operation. The proposed method involves conversion of received residues into mixed radix form. Based on the error detecting capability of RRNS code under study, moduli projection is carried out iteratively to determine the corresponding integers. The proposed algorithm considerably reduces the number of required iterations involved in the recovery process, therefore improving the speed of decoding operation.

Keywords: Modified modulus projection algorithm, RRNS, Signal processing, Error control, Radix

INTRODUCTION

A Residue Number System (RNS) for integer is a method of expressing Weighted Number System (WNS) by a set of its remainders when divided by set of n pairwise positive integers. To realize error-control scheme, this set of remainders, called residues is appended with extra ones. These additional residues constitute parity digits, hence the term Redundant Residue Number System (RRNS). Parallelism, modularity, fault-tolerance, carry-free operations, and lack of order significance among the residues are some of features that make RNS-based error correcting schemes very attractive (Yang and Hanzo, 2001). Notable efforts have been made with respect to RRNS codes by many scholars. Among these are self-checking architectures in RRNS (Hastings and Watson, 1966); development

of concept of legitimate and illegitimate ranges (Barsi and Maestrini, 1973); application of Mixed Radix Conversion (MRC) and Base Extension (BEX) algorithms in RRNS based digital filters and residue number error checkers by Jenkins (1983), and Altman and Jenkins (1988); development of coding theoretic framework for RRNS Krishna *et al.* (1992), and Krishna and Sun (1992); a method of multiple error-controls in RRNS codes employing Chinese Remainder Theorem (Goh and Siddiqi, 2008); unified framework on RRNS for computations of both integer and polynomial (Khonji *et al.* 2010); systematic RRNS codes performance over Additive White Gaussian Noise (AWGN) and Rayleigh channels utilizing Chase algorithm (Liew *et al.*, 2006) and RRNS code for fault-tolerant hybrid memories (Haron and Hamdioui, 2011).

This work is based on the strategy where estimates of transmitted integer are computed from the received residues as done in the work presented by Mandelbaum (1973), and Goh and Siddiqi (2008). In contrast to these works, the proposed scheme here neither requires complex optimization nor involve large modulo operation because CRT is not employed. Briefly, the proposed scheme involves conversion of the received residues to its mixed radix form. If all redundant mixed radix digits are zeros, then the received residues is error free and the integer is computed from mixed radix digits via BEX operation. If some or all of redundant mixed radix digits are non-zero, then the received residues are declared to be erroneous. Based on the error detecting capability of the RRNS code in question, the process of moduli projection is performed iteratively to determine the corresponding integer. The Hamming distances are computed from residue representation of integers found lying within the legitimate range and the received residue. Any of the integers that have Hamming distance of less or equal to the error correcting capability of the RRNS code is the transmitted integer.

REDUNDANT RESIDUE NUMBER SYSTEM

Because RRNS is a number system, its range of validity has to be specified. Thus, an RRNS is defined on a set of n pair-wise positive integers $m_1, m_2, \dots, m_k, m_{k+1}, \dots, m_n$ that is regarded as moduli (Yang and Hanzo, 2001). Among these n moduli, the first k moduli are called non-redundant moduli set. The product of this set of n pair-wise positive integers defined the dynamic

range M of the RNS. The dynamic range is given by equation (1):

$$M = \prod_{i=1}^k m_i \quad (1)$$

The remaining $r = n - k$ moduli constitute redundant moduli set. This set enables error detection and correction in the RRNS code.

Let M_R be the product of redundant moduli which is defined as equation (2):

$$M_R = \prod_{i=k+1}^n m_i \quad (2)$$

Every integer X in the range $(M + \mu)$

where μ could zero or positive integer.

X can be expressed by a unique residue sequence X having n components

x_1, x_2, \dots, x_n . This unique residue sequence in compact form can be written as equation (3):

$$X \Leftrightarrow x_{i=1}^n m_i \quad (3)$$

$$\text{where } x_i \equiv X \pmod{m_i} \quad (4)$$

$A \equiv B$ means A is congruent to B and the

integer x_i is the i^{th} residue digit of the modulus m_i that satisfies $0 \leq x_i < m_i$.

To obtain the original number in WNS from its set of residues, two prominent approach-

es are often used in the literatures. These are CRT and combination of BEX with MRC.

According to CRT, for any given residue

vector $(x_1, x_2, x_3, \dots, x_n)$ where $0 \leq x_i < m_i$ for $i = 1, 2, 3, \dots, n$, there exist one and

only one integer X such that $0 \leq X < M$

and $x_i \equiv X \pmod{m_i}$. X is evaluated by equation (5):

$$X \equiv x_i T'_i M'_i \pmod{M} \quad (5)$$

$$\text{Where } M'_i = \frac{M}{m_i} \quad (6)$$

The integers $T'_i, i = 1, 2, 3, \dots, n$ are determined a priori using the congruencies given in equation (7):

$$T'_i M'_i \equiv 1 \pmod{m_i} \quad (7)$$

The integers T'_i are the multiplicative in-

$$a_n = \left(\left(\left(\left(\left(x_n - a_1 \right) m_1^{-1} - a_2 \right) m_2^{-1} - \dots - a_{n-1} \right) m_{n-1}^{-1} \right) m_n \right)$$

In a RRNS, the digits a_1, a_2, \dots, a_k are regarded as the non-redundant MRDs or

simply, information digits, while $a_{k+1}, a_{k+2}, \dots, a_n$

are referred to as the parity MRDs or redundant digits. For the MRDs satisfying

$0 \leq a_i < m_i$, any positive number in the

verses of M'_i .

The major drawback of this method large is the modular operation that is involved; this tends to make real-time implementation of the CRT impractical. In order to avoid processing of large integers for the recovery of integer from its residue the alternative method is the use of BEX alongside the MRC method [1]. This is formulated as follows:

Given a set of residues $(x_1, x_2, x_3, \dots, x_n)$ defined on the corresponding set of moduli

$[m_1, m_2, \dots, m_n]$ and a set of Mixed Radix

Digits (MRD) (a_1, a_2, \dots, a_n) the decimal equivalent of the residues can be determined from equations (8) and (9):

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2 + \dots + a_n \prod_{i=1}^{n-1} m_i \quad (8)$$

where MRD are:

$$a_1 = x_1$$

$$a_2 = ((x_1 - a_1) m_1^{-1})_{m_2}$$

range $[0, \prod_{i=1}^n m_{i-1})$ can be uniquely represented.

ORIGINAL ALGORITHM

The algorithm for locating a single residue digit error that is based on the properties of modulus projection and MRC was presented in many of previous works such as [3] -[5],

[13] and [14]. Going by these works, the modulus m_i - projection of X in an RRNS (n, k) code, given by X_i , can be written as equation (10):

$$X_i \equiv X \left(\text{mod} \frac{MM_R}{m_i} \right) \quad (10)$$

From equation (10), it implies that X_i can be expressed as $(X_1, X_2, \dots, X_{i-1}, X_{i+1}, \dots, X_n)$, which is a reduced residue form representation of X since i^{th} residue digit X_i is deleted. The mixed radix representation of X_i admits the form of equation (11):

$$X_i = \sum_{l=1}^n a_l \prod_{r=1}^{l-1} m_r \quad l \neq i, r \neq i \quad (11)$$

The first k digits of equation (11) are still

Where $M_v = \prod_{k=1}^{\sigma} m_{i_k}$, $v = \{i_1, i_2, \dots, i_c, \dots, i_{\sigma}\}$, $i_1 < i_2 < \dots < i_c < \dots < i_{\sigma}$ and $\sigma \leq n - k = d - 1$.

The minimum code distance is defined as d . The MRD representation of X_v can be expressed as equation (13):

$$X_v = \sum_{l=1}^n a_l \prod_{r=1}^{l-1} m_r \quad l \neq v, r \neq v \quad (13)$$

The legitimate and illegitimate range of the resulting reduced RRNS are $[0, M')$ and

regarded as the non-redundant MRDs and the remaining as parity MRDs. It is clear that

if X_i is a legitimate number, all of parity MRDs will be zero. Claudio and Piazza [13] established that an illegitimate projection due to a single residue digit error cannot be smaller than the smallest nonzero number given by equation (8) with the entire parity MRDs equal zero. Thus, algorithm for a single residue digit error control only involves checking whether the parity MRDs is zero

for each modulus m_i - projection $i = 1, 2, 3, \dots, n$.

Krishna et al.[6] extended this idea to accommodate more than one residue digits error control in RRNS code.

Let X_v be the M_v - projection of an integer X , which is given by equation (12):

$$X_v \equiv X \left(\text{mod} \frac{MM_R}{M_v} \right) \quad (12)$$

$$\left[M', \frac{MM_R}{M_v} \right] \text{ respectively. The product of the first } k \text{ moduli of the reduced RRNS gives } M' \text{ that is expressed as equation (14):}$$

$$M' = \prod_{r=1}^{k+c} m_r, \quad i_c \leq k+c, i_{c+1} > k+c, i_1 \leq k, r \neq v$$

$$M' = M = \prod_{r=1}^k m_r, \quad i_1 > k \quad (14)$$

It is clear that $M' \geq M$. The value of M' is equal to the smallest nonzero number represented by equation (10) with all parity MRDs equal zero. It is obvious from equation (13) that the M_v -projection of a legitimate number X in RRNS is also legitimate in $[0, M)$, implying $X_v = X$. Krishna et al [6] summed this up in a theorem that is described as follow, suppose in a RRNS (n, k) code \bar{X} is an illegitimate integer message that differs from another integer message X that is legitimate in the $i_{1th}, i_{2th}, \dots, i_{\sigma th}$ residue digits, where $\sigma \leq (n-k)$, then the M_v -projection X_v is a legitimate integer.

The theorem above suggests that the correct or (transmitted) integer message X can be recovered from an illegitimate (or erroneously received) integer message by drop-

ping some of the received residue digits and their corresponding moduli, provided that the dropped residue digits are erroneous ones and that the reduced RRNS exhibits a sufficiently high dynamic range to unambiguously represent the integer constituting the message. Thus using this theorem, it can be determined whether a projection is legitimate or not by checking only the parity MRDs. If all parity MRDs are zero, the projection is legitimate; otherwise the projection is illegitimate.

Suppose we have n -digit residues in a RRNS (n, k) code that has error correcting capability t with the set of received residues given by y . By applying moduli projection approach, it will take at most p number of trials $(p = {}^nC_t)$ to arrive at correct error positions where n and t are respectively the number of residue digits and the error correcting capability of the RRNS (n, k) code.

Going by the above, the algorithm for decoding RRNS (n, k) code can be highlighted as follows:

Step 1: Get the received residue vector y and the corresponding moduli m_i on which the RRNS (n, k) code is defined.

Step 2: Compute the MRDs based on y and m_i . Check if all MRDs are zero. If yes, declare y as error free, compute the transmitted WNS integer from the residues using equations (8) and (9). Otherwise, proceed to step 3.

Step 3: Based on error correcting capability

of the RRNS (n, k) code, perform modulus projection on \mathcal{Y} to obtain reduced form of the RRNS (n, k) code.

Step 4: In each case, compute the MRDs and the WNS equivalent using equations (8) and (9)

Step 5: The integer found from step 4 that has all its parity MRDs equal zero is the correct integer that is sent.

PROPOSED ALGORITHM

A RRNS (n, k) code with higher error correcting capability requires large number of residue digits. Consequently, the number of trials p for determination of error positions also increases, thus increase in the computation overhead. To address this problem, there is a need for modification of the above algorithm so as to lessen the computational overhead of the error correc-

tion scheme. Given a RRNS (n, k) code with $t(= \lfloor n-k \rfloor / 2)$ error correcting capa-

bility, if we select $\beta(= n-k)$ error positions such that most of the possible combinations are taken care of at a time, a total of

$h = \beta C_t$ combinations of t errors are covered in a single iteration. This can be utilized to evolved decoding process with reduced number of required iterations [15]. The idea is better explain with the aid of numerical illustrations.

Consider cases of the following RRNS

(n, k) codes: (8, 3) code; (11, 5) code and (12, 3) code. The error correcting capability

is $t(= \lfloor n-k \rfloor / 2)$. The first two codes have error correcting capability of 2 as $t=2, \beta=4$. The next two can correct

RRNS (8, 4) code: (1, 2, 3, 4); (3, 4, 5, 6); (5, 6, 7, 8); (7, 8, 1, 2) i.e. 4 alternatives

RRNS (9, 5) code: (1, 2, 3, 4); (3, 4, 5, 6); (5, 6, 7, 8); (7, 8, 9, 1); (9, 1, 2, 3); (2, 3, 4, 5); (4, 5, 6, 7); (6, 7, 8, 9);

(8, 9, 1, 2) i.e. 9 alternatives

RRNS (9, 3) code: (1, 2, 3, 4, 5, 6); (4, 5, 6, 7, 8, 9); (7, 8, 9, 1, 2, 3) i.e. 3 alternatives

RRNS (10, 4) code: (1, 2, 3, 4, 5, 6); (4, 5, 6, 7, 8, 9); (7, 8, 9, 10, 1, 2); (10, 1, 2, 3, 4, 5); (3, 4, 5, 6, 7, 8);

(6, 7, 8, 9, 10, 1); (9, 10, 1, 2, 3, 4); (2, 3, 4, 5, 6, 7); (5, 6, 7, 8, 9, 10); (8, 9, 10, 1, 2, 3)

i.e. 10 alternatives

RRNS (11, 3) code: (1, 2, 3, 4, 5, 6, 7, 8); (5, 6, 7, 8, 9, 10, 11, 1); (9, 10, 11, 1, 2, 3, 4, 5); (2, 3, 4, 5, 6, 7, 8, 9);

(6, 7, 8, 9, 10, 11, 1, 2); (10, 11, 1, 2, 3, 4, 5, 6); (3, 4, 5, 6, 7, 8, 9, 10); (7, 8, 9, 10, 11, 1, 2, 3);

(11, 1, 2, 3, 4, 5, 6, 7); (4, 5, 6, 7, 8, 9, 10, 11); (8, 9, 10, 11, 1, 2, 3, 4) i.e. 11 alternatives

RRNS (12, 3) code: (1, 2, 3, 4, 5, 6, 7, 8, 9); (5, 6, 7, 8, 9, 10, 11, 12, 1); (9, 10, 11, 12, 1, 2, 3, 4, 5)

i.e. 3 alternatives

RRNS (12, 4) code: (1, 2, 3, 4, 5, 6, 7, 8); (5, 6, 7, 8, 9, 10, 11, 12); (9, 10, 11, 12, 1, 2, 3, 4)

i.e. 3 alternatives

The following observations can be deduced from above:

- i. Possible error position selections are function of error detecting capability of the code;
- ii. Any two possible error position selections differ by number of digit that equal the error correcting capability of the code;
- iii. When the value of n is a multiple of the value of k , total possible combinations is less or equal the value of k ;
- iv. When the value of n is not a multiple of k , total possible combinations are less or equal the value of n .

Thus, the required number of iterative cycles required in the recovery of the original integer using the proposed method is given by equation (15) as:

$$h_{max} = n \quad (15)$$

Invocation of modulus projection algorithm on the received residue digits sequence using the above combinations of possible error positions is the next step. This will yield more than one possible solutions falling within the legitimate range. This is due to the fact that the algorithm attempt correction of residue digits errors greater than the code capability. To determine the correct transmitted integer from the resulting solutions falling within the legitimate range, Maximum Likelihood Decoding (MLD) [16] approach is used.

Suppose the set of solutions that fall within the legitimate range obtained from a scheme set to correct $\beta > t$ residue digits

error in an RRNS (n, k) code is given by equation (16):

$$Z = \{Z_1, Z_2, \dots, Z_{i+1}, \dots, Z_\tau\} \quad (16)$$

where τ is the number of solution that falls within the legitimate range of the RNS. For each of $Z_j, j=1, 2, 3, \dots, \tau$ the correspond-

ing residue vector r_j is obtained using equation (4). Applying MLD, the Hamming dis-

tance $d(y, z_j)$ between the received residue vector y and each of the computed residue

vector r_j is evaluated. In line with the MLD any of the received residue vector that has Hamming distance less than or equal to t (the error correcting capability of the code),

its corresponding integer Z_j is the transmitted integer.

The proposed algorithm for decoding of RRNS (n, k) code could be summarized as follows:

Step 1: obtain the received residue vector y and the corresponding moduli m_i on which the RRNS is defined.

Step 2: compute the MRDs based on y and m_i . Check if all parity digits are zero. If yes, declare the received residues as error free,

compute Y using equations (8) and (9) and stop. Otherwise, proceed to the next step 3.

Step 3: Determine the projection depth $\beta = (n - k)$ and perform moduli projection of depth β on the received residue y for

$c=1$ to n (the maximum number of possible selections) to get reduced residue y_c .

Step 4: While $c \leq n$, compute estimate \bar{y}_c of the integer from y_c for each selection using equations (8) and (9). If \bar{y}_c is in the legitimate range, set $\bar{y}_c = Z_j$ and proceed to Step 5. Otherwise, increase c by 1.

Step 5: Calculate the residue vector r_j from R_j and the Hamming distance $d(y, z_j)$.

Step 6: Output $X = R_j : d(y_c, z_j) \leq t$

SIMULATION RESULTS

To demonstrate multiple errors correcting capability of the algorithm, the following examples are considered.

Example 1-Two residue error correction with Original Algorithm

Consider an RRNS (7, 3) code. The code can correct up to $t = 2$ residue digit errors. The chosen moduli set are {3, 5, 7, 8, 11, 13, 17},

the legitimate range of

the code is [0, 105) and the illegitimate range is [105, 2042040). Suppose the transmitted

integer message $X = 89$, $x \equiv (2, 4, 5, 1, 1, 11, 4)$ when coded into residue vector. Also, suppose the received vector has two errors introduced in the course of transmission at

positions $u_1 = 3$ and $u_2 = 4$ so that the received vector is $y \equiv (2, 4, 6, 7, 1, 11, 4)$.

Using y and m_i , mixed radix digits are computed to have these values $a_4 = 7$, $a_5 =$

5, $a_6 = 12$ and $a_7 = 13$, which are all non-zero. Thus, confirming existence of errors in the received residue vector. In line with the algorithm, modulus projections of all combi-

nations of $m_i m_j$ are carried out. In each case of resulting reduced form of RRNS code, MRDs and equivalent WNS are computed. The results are presented in Table 1. In line with the algorithm, the only valid result from Table 1 where all parity MRDs are

zeros is $X = 89$.

With this result the positions of errors are determined to be $u_1 = 3$ and $u_2 = 4$. The algorithm has correctly determined the transmitted integer after 21 iterations.

Table 1: Result of multiple error Correction Algorithm with p=21 Iterations

Error Positions	Mixed Radix Digits						Estimates
e_1	e_2	a_1	a_2	a_3	a_4	a_5	\bar{Y}
1	2	6	3	7	12	13	419
1	3	4	3	10	9	12	419
1	4	4	6	0	2	12	34
1	5	4	6	3	4	10	139
1	6	4	6	3	1	9	139
1	7	4	6	3	1	5	139
2	3	2	3	6	3	4	155
2	4	2	6	8	7	14	188
2	5	2	6	3	7	11	83
2	6	2	6	3	2	15	83
2	7	2	6	3	2	4	83
3	4	2	4	5	0	0	89
3	5	2	4	3	10	12	59
3	6	2	4	3	3	4	59
3	7	2	4	3	3	3	59
4	5	2	4	2	6	2	44
4	6	2	4	6	3	7	104
4	7	2	4	6	3	9	104
5	6	2	4	6	3	16	104
5	7	2	4	6	3	1	104
6	7	2	4	6	3	0	104

Example 2-Two residue digit error correction with Modified Algorithm

Modified algorithm is applied to the same RRNS (7, 3) code. Since all redundant MRDs are non-zero, projection depth is determined to be $\beta = n - k = 4$. Moduli projection of depth β is carried out on the received residue. In each case, estimate of

equivalent WNS is determined. Table 2 presented results obtained. From Table 2, three of possible solutions fall within the legitimate range with one occurring in triplicates.

$$Z = (79; 89; 104)$$

Let them be z_i . To determine the actual transmitted integer from the

three, the residue vectors z_i and Hamming distances from the received residue vector

y are calculated for each in line with algorithm. The result is presented in Table 3.

From Table 3, the only value of z_i that has a Hamming distance $\leq t=2$ is 89. Thus, the proposed algorithm has recovered the correct transmitted integer after 7 iterations.

Table 2: Result of double residue digits errors correction with modified Algorithm, $h=7$

H	Error Positions				Mixed Radix Digits			\bar{Y}
	e_1	e_2	e_3	e_4	a_1	a_2	a_3	
1	1	2	3	4	1	8	0	89
2	3	4	5	6	2	4	5	89
3	5	6	7	1	4	1	2	79
4	7	1	2	3	7	2	4	375
5	2	3	4	5	2	3	2	89
6	4	5	6	7	2	4	6	104
7	6	7	1	2	6	7	1	111

Table 3: The residues and Hamming distances for two residues digits error correction

i	Z	z_i	y	$d(z_i, y)$
1	79	1, 4, 2, 7, 2, 1, 11	2, 4, 6, 7, 1, 11, 4	5
2	89	2, 4, 5, 1, 1, 11, 4	2, 4, 6, 7, 1, 11, 4	2
3	104	2, 4, 6, 0, 5, 0, 2	2, 4, 6, 7, 1, 11, 4	4

Example 3 - Three residue digit errors correction using Modified Algorithm

RRNS (9, 3) code is used, the moduli set employed are {23, 25, 27, 29, 31, 32, 37, 39, 43}

and the legitimate range is [0, 15525). The chosen integer message is 14973. The encoded residue equivalent is $X \equiv (0, 23, 15, 9, 0, 29, 25, 36, 9)$

. Suppose the received residue vector is

$$y \equiv (2, 23, 15, 10, 3, 29, 25, 36, 9)$$

i.e. residue

digit positions 1, 4 and 5 are erroneous. Six possible error positions are selected because the projection depth is 6 for application of modified algorithm. Table 4 gives results obtained. Three possible solutions that fall within the legitimate range are represented as $Z = \{14793, 3498, 5955\}$

. In line with the

algorithm, the residue vectors z_i as well as

their Hamming distances from the received residue vector \bar{y} are calculated. The result is presented in Table 5. As can be seen in Table 4, the only value of z_i that has a Hamming distance which is

less than or equal to $t = 3$ is 14793. Therefore, the modified algorithm has again correctly recovered the transmitted integer with a reduced number of iterative processes, which in this case equal 3.

Table 4: Result of three residue digits errors correction with modified Algorithm, $h=3$

H	Error Positions						Mixed Radix Digits			\bar{y}
	e_1	e_2	e_3	e_4	e_5	e_6	a_1	a_2	a_3	
1	1	2	3	4	5	6	25	14	10	14793
2	4	5	6	7	8	9	2	2	6	3498
3	7	8	9	1	2	3	10	19	6	5955

Table 5: The residues and Hamming distances for three residue digits error correction

i	z	z_i	y	$d(z_i, y)$
1	14793	0, 23, 15, 9, 0, 29, 25, 36, 9	2, 23, 15, 10, 3, 29, 25, 36, 9	3
2	3498	2, 23, 15, 18, 26, 10, 20, 27, 15	2, 23, 15, 10, 3, 29, 25, 36, 9	6
3	5955	21, 5, 15, 10, 3, 3, 35, 27, 21	2, 23, 15, 10, 3, 29, 25, 36, 9	6

DISCUSSION

Original and modified algorithms were used to recover the correct encoded integer in RRNS (7, 3) code respectively in examples 1 and 2. The original algorithm recovered the transmitted integer with higher number of iterations while the proposed algorithm employed lesser number of iterations in the recovery process. Original algorithm required twenty-one iterations while the modified algorithm only takes one-third of that i.e. seven iterations. A huge saves in the computation overhead is realized with the

use of the proposed algorithm. Similarly, in example three the number of iterations involved in the recovery process using modified algorithm is three whereas the number would have been eight-four if the original algorithm is employed.

CONCLUSION

In this paper, the property of modulus projection and mixed radix conversion technique has been employed in developing efficient decoding algorithm for RRNS codes. The developed algorithm is suitable for cor-

rection of an arbitrary number of residue digit errors in RRNS codes. To ensure this, two things are required. First, the number of residue digits in errors must not be greater than the error correcting capability of the RRNS code. Two, the depth of moduli-projection must be equal to the number of redundant moduli in the RRNS code. The developed modified modulus projection algorithm required minimal number of iterations in the recovery process, which led to considerable reduction in the computation overhead.

REFERENCES

- Altman E.J., W.K. Jenkins** .1988. Self-Checking Properties of Residue Number Error Checkers Based on Mixed Radix Conversion. *IEEE Trans. Circuits Syst.*, 159-167.
- Amusa K.A., Nwoye E. O.** 2012. Novel Algorithm for Decoding Redundant Residue Number Systems Code. *IJRRAS* 12 (1): 158 – 163.
- Barsi F., Maestrini, P.** 1973. Error Correcting Properties of RRNS. *IEEE Trans. Compt.*, 307-315
- Claudio E. D., Piazza, F.** .1983. A systolic Redundant Residue error correction Circuit. *IEEE Trans. Compt.*, 42(4): 427-432.
- Goh V. T., Siddiqi, M.U.** 2008. Multiple Errors Detection and Correction Based on RRNS. *IEEE Trans. Comm.* 56(3): 325-330.
- Hastings C. W., Watson, R.W** 1966. Self-checked Computations using Residue Arithmetic. *Proceeding of IEEE*, 54(12): 1920 – 1931.
- Hiasat A.A.** 2003. Efficient Residue to Binary Converter. *IEEE Proceeding Digital Technology*
- Jenkins W. K.** 1983. The Design of error checkers for Self-checking Residue Number Arithmetic. *IEEE Trans Computer*, 388-396.
- Krishna H., Lin, K.Y., Sun, J.D.** 1992. A Coding Theory Approach to Error Control in RRNS-part 1:-Theory and Single error-Correction. *IEEE Trans. Circuits and Syst. II Analog & DSP*, 39: 8- 17.
- Krishna H., Sun, J.D.** 1992. A Coding Theory Approach to Error Control in RRNS-part II: Multiple errors Detection and Correction. *IEEE Trans. Circuits and Syst. II Analog & DSP*, 39:18-34.
- Liew T. H., Lie-Liang Yang, Lajos Hanzo.** 2006. Systematic RRNS Codes: Analytical Upper Bound and Iterative Decoding Performance over AWGN and Rayleigh Channels. *IEEE Trans. Communications*, 54(6): 1006-1016
- Liu Y. C., Yau, S.S.** 1973. Error Correction in RRNS. *IEEE Trans. Compt.*, C-22: 5-11.
- Majid Khonji, Clement Pernet, Jean-Louis Roch, Thomas Roche., Thomas Stalinski** 2010. Output sensitive Decoding for RRNS. *200X ACM X-XXXXX-XX-X/XXX/XX*
- Mandelbaum D. M.** 1976. On a class of Arithmetic Codes and a Decoding Algorithm. *IEEE Trans. Information Theory*, 22: 85-88.
- Nor Zaidi Haron., Said Hamdioui** 2011. RRNS Code for fault-tolerant hybrid memories. *ACM Journal Emerging Technology in Computing Systems*, 7(1), Article 4.

Yang L.L., Hanzo, L. 2001. RRNS Based [Technol. Conf.](#), 1472-1476.
Error Correction Codes. [Proc. 54th Vehicle](#)

(Manuscript received: 18th August, 2015; accepted: 7th June, 2016)