Multiple Bits Error Detection and Correction in RRNS Architecture using the MRC and HD Techniques

Yaw Afriyie
Department of Accounting
School of Business and Law
University for Development Studies
Wa, Ghana

M. I. Daabo
Department of Computer Science
Faculty of Mathematical Sciences
Universityf for Development Studies
Navrongo, Ghana

ABSTRACT

Transferring data between two points is very essential in digital systems and the accuracy of the transferred data is important for some critical applications. However, errors during the transmission of data are very common in these systems. RRNS is mostly used in parallel processing environments because of its ability to increase the robustness of information passing between computer processors. This paper presents some results on multiple error detection and correction based on the Redundant Residue Number System (RRNS). The Mixed Radix Conversion (MRC) was incorporated with the Hamming Distance (HD) as a joint technique in the detection and correction of multiple bits errors. In the proposed method, it is possible to detect the exact locations of multiple bits errors and correct them using minimum hardware. An area-delay comparison analysis was done and compared with the other best-known result, which revealed that, the proposed scheme has a considerable improvement in speed by up to 68% and tends to require about 81% less hardware resources, which proves the efficiency of the proposed scheme in terms of delay and area requirements.

General Terms

Residue Number System, Redundant Residue Number System (RRNS) Mixed Radix Conversion, Hamming Distance, Error Detection and Correction, Digital Signal Processing, Computer Architecture.

Keywords

MRC, Mixed Radix Digits, Residue Number System (RNS), Redundant Residue Number System (RRNS), Hamming Distance (HD).

1. INTRODUCTION

Residue Number System (RNS) is an unconventional, high speed, and fault tolerance number system that has many applications in digital computing such as image processing systems[1], digital signal processing (DSP) [2], RSA algorithm [3], and digital communications [4]. Moreover, RNS is a useful tool for implementation of high speed FIR filters [5]. In RNS, instead of sending a number, using modulus in the moduli set, the remainders of the numbers are sent and all of the operations are performed in parallel.

Therefore, its operations result in very high speed. Moreover, the modulus in the residue set act as secret keys, then converts back the received remainders to the original number in the modulus set. In addition to these, by adding some redundant modulus, Redundant RNS (RRNS) is constructed that has error control ability. Because reliability in data delivery is a critical issue in many applications, fault tolerance is a remarkable feature of RRNS. A coding theoretical approach to error control coding invoking the RRNS has been developed by Krishna et.al [6]. RNS offers better speed because of its underlying carry-free property. Because of this, a number of algorithms have been proposed to detect and correct errors in RRNS. In addition, the concepts of Hamming weight, minimum distance, weight distribution, error detection capabilities, and error correction capabilities are investigated. Goh and Siddiqui [9] proposed an algorithm that detects and corrects multiple bits errors in RNS using the Chinese Remainder Theorem (CRT). The CRT uses large numbers in its computations which results in reducing the performance of the algorithm. This is similar to the algorithms proposed by Sun & Krishna [7] and Yang & Hanzo [8]. Their proposed schemes however utilizes the MRC and syndrome check with the help of look-up table. The effect of this in terms of hardware resources and memory are more expensive when compared with proposed scheme. In addition, the proposed algorithm is straightforward and easier to implement. The proposed algorithm is also less computationally intensive and makes it more efficient in the detection and correction of multiple bit errors than the schemes in Sun & Krishna [7] and Yang & Hanzo [8]. Amusa & Nwoye [10] described a computationally efficient decoding procedure relying on the projection-depth theory for correcting multiple errors. The proposed scheme will eliminate multiple bit errors in RRNS by using other schemes like the MRC and the HD as a joint technique with lower complexity compared to higher order of complexity of CRT by detecting and correcting the errors. Because reliability in data transmission is a critical issue in digital systems, fault tolerance is an important feature of redundant residue number systems.

In this paper, we proposed a new scheme that will show the effectiveness of the RRNS based on MRC and the HD that will detect and correct multiple bit errors in RRNS. The proposed scheme outperforms best known similar state-of-the-art error detection and correction schemes.

2. RESIDUE ARITHMETIC FUNDAMENTALS

RNS is characterized by a set of k pairwise relatively prime positive integers, i.e. the greatest common divisor gcd $(\mathrm{m_i}, \mathrm{m_j})$ =1 with $\mathrm{i} \neq \mathrm{j}$, $\mathrm{m_1}, \mathrm{m_2} \ldots \mathrm{m_{k-1}}, \mathrm{m_k}$ called the moduli, that is formed in increasing, i.e., $\mathrm{m_1} < m_2 < \cdots < \mathrm{m_{k-1}} < m_\mathrm{k}$[12] - [14]. Their products represent the interval [0, M) called the legitimate range that defines the useful computational range of the number system, that is,

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

To represent positive and negative numbers, the dynamic range is defined as $[-(\mathrm{M}-1)/2\,,(\mathrm{M}-1)/2]$ if M is odd and $\mathrm{M}/2$ if M is even. Every natural integer X in the legitimate range can be represented by a set of residues $\mathrm{r}_1, \mathrm{r}_2\,\ldots,\mathrm{r}_{\mathrm{k}-1},\ \mathrm{r}_\mathrm{k}$ where

$$\mathrm{r}_\mathrm{i} \equiv \mathrm{X}(\mathrm{mod})\mathrm{m}_\mathrm{i} \qquad (2)$$

With i ϵ [1, k] and $|\mathrm{X}\ |\mathrm{m}_\mathrm{i}$ denotes X modulo m_i. Due to the carry-free property, the three operations namely addition, subtraction and multiplication can be operated with respect to the moduli independently, i.e.

$$\mathrm{x}_1, \mathrm{x}_2 \ldots \mathrm{x}_\mathrm{k}\ *\ \ \mathrm{x}_\mathrm{k} * \mathrm{y}_1, \mathrm{y}_2 \ldots \mathrm{y}_\mathrm{k} = \mathrm{z}_1, \mathrm{z}_2\ \ \ldots\ \ \mathrm{z}_\mathrm{k},\ \mathrm{z}_\mathrm{i} \equiv |\mathrm{x}_\mathrm{i} * \mathrm{y}_\mathrm{i}\ |\mathrm{m}_\mathrm{i} \qquad (3)$$

With * denotes the three operations. Consequently, RNS is able to provide a fast arithmetic.

Redundant Residue Number System (RRNS) is achieved by adding some redundancy to the Residue Number System. RRNS helps in both error detection and error correction. By adding $(\mathrm{u}-\mathrm{w})$ redundant moduli $(\mathrm{m}_{\mathrm{n}+1}, \mathrm{m}_{\mathrm{n}+2}, \ldots, \mathrm{m}_\mathrm{n})$ to the v information moduli $(\mathrm{m}_1, \mathrm{m}_2, \mathrm{m}_3, \ldots, \mathrm{m}_\mathrm{n})$, a RRNS (u, w) code can be generated. This process is called RRNS encoding. Thus, an integer X is represented in the RRNS form as

$$\mathrm{X} = \{\mathrm{r}_1, \mathrm{r}_2, \mathrm{r}_3, \ldots, \mathrm{r}_\mathrm{w}, \mathrm{r}_{\mathrm{w}+1}, \ldots, \mathrm{r}_\mathrm{u}\} \qquad (4)$$

where $(\mathrm{m}_1, \mathrm{m}_2, \mathrm{m}_3, \ldots, \mathrm{m}_\mathrm{n})$ are called information moduli and $(\mathrm{m}_{\mathrm{w}+1}, \mathrm{m}_{\mathrm{w}+2}, \mathrm{m}_{\mathrm{w}+3}, \ldots, \mathrm{m}_\mathrm{w})$ are called the redundant moduli. Similarly, $(\mathrm{r}_1, \mathrm{r}_2, \mathrm{r}_3, \ldots, \mathrm{r}_\mathrm{w})$ are called the information residues and $(\mathrm{r}_{\mathrm{w}+1}, \mathrm{r}_{\mathrm{w}+2}, \mathrm{r}_{\mathrm{w}+3}, \ldots, \mathrm{r}_\mathrm{w})$ are called the redundant residues.

For RRNS, even if some of the redundant residues are removed, an integer can be recovered if the retained residue digits are correct.

Theorem 1: RRNS $(\mathrm{u}-\mathrm{w}, \mathrm{v})$ code has a detection capability of $(\mathrm{u}-\mathrm{w}-\mathrm{v})$ errors and an error correction capability of $(\mathrm{u}-\mathrm{w}-\mathrm{v})/2$ [10]. The code rate of a redundant residue number system can be defined as

$$\mathrm{R}_\mathrm{C} = \frac{\mathrm{K}_\mathrm{b}}{\sum_{\mathrm{j}=1}^{\mathrm{u}} \mathrm{K}_\mathrm{b}} \qquad (5)$$

where $\mathrm{K}_\mathrm{b} = \lfloor \log_2 \mathrm{M}_\mathrm{r} \rfloor$ and $\mathrm{K}_{\mathrm{bj}} = \lceil \log_2 \mathrm{m}_\mathrm{j} \rceil$, where $(\mathrm{j} = 1, 2, 3, \ldots, \mathrm{u})$ are the moduli. By varying the number of redundant bits that are transmitted, the code rate and error correction capability are also varied. Redundancy is added to the information bits, therefore the code rate decreases and error correction property is improved. In RNS, the number of non-zero elements in a vector is defined as its hamming weight. Let Y_i and Y_j be two codevectors, then the hamming distance $\mathrm{d}(\mathrm{Y}_\mathrm{i}, \mathrm{Y}_\mathrm{j})$ is defined as the number of bits in which two codevectors Y_i and Y_j differ. Minimum distance, d is the minimum of the hamming distances

$$\mathrm{d} = \min\big(\mathrm{d}(\mathrm{y}_{\mathrm{i},}\mathrm{y}_\mathrm{j}); \mathrm{y}_\mathrm{i} \neq \mathrm{Y}_\mathrm{j}\big). \qquad (6)$$

Theorem 2: The minimum hamming distance (d) of an $\mathrm{RRNS}(\mathrm{u}, \mathrm{w})$ code is defined as $\mathrm{d} = \mathrm{u} - \mathrm{w} + 1,$ provided $(\mathrm{m}_1, < \mathrm{m}_2 < \mathrm{m}_3 < \cdots < \mathrm{m}_\mathrm{u})$.

Theorem 3: the minimum distance of an RRNS code is $\mathrm{d}_{\mathrm{min},}$ if and only if the product of redundant moduli satisfies these relations:

$$max\left\{\prod_{i=1}^{d_{min}} m_{ji}\right\} > M_{n-k}$$
$$\geq max\left\{\prod_{i=1}^{d_{min-1}} m_{ji}\right\} \qquad (7)$$

where $\mathrm{M}_{\mathrm{n}-\mathrm{k}} = \prod_{\mathrm{j}=\mathrm{k}+1}^{\mathrm{n}} \mathrm{m}_\mathrm{j}$ represents the product of the redundant moduli of the code and m_{ji} is any of the n moduli of the RRNS code, for $1 \leq \mathrm{j}_\mathrm{i} \leq \mathrm{n}$.

Theorem 4: A code y, based on a redundant residue number system has a minimum nonzero hamming weight $\mathrm{wt}_{\min} \geq \mathrm{r}+1$ and a minimum distance $\mathrm{d}_{\min} = \mathrm{r}+1$ [13].

Theorem 5: For a redundant residue number system, the error detecting capability (c), $\mathrm{c} = \mathrm{d}-1$ and the error correcting capability $\mathrm{t} = \left\lfloor \frac{\mathrm{d}-1}{2} \right\rfloor$ where $\lfloor \mathrm{x} \rfloor$ is the largest integer value smaller than x[11]. Thus, RRNS (u, w) code can detect up to $(\mathrm{u}-\mathrm{w})$ residue digits and correct up to $\mathrm{t} = \left\lfloor \frac{\mathrm{u}-\mathrm{w}}{2} \right\rfloor$ residue digits. This implies that, single error and multiple error detection and correction algorithms can be developed by suitably selecting u and w. This paper focuses on the detection and correction of multiple bit errors in RRNS with $\mathrm{u}-\mathrm{w} = 2$.

2.1 Conversion

It is well known that MRC and CRT are approaches that are often applied in conversion. This can be seen in the works of [3]. This study will be limited to the MRC and the HD techniques because the real time implementation of the CRT involves a modular operation with a large integer M which results in large complexities. To prevent the computations with such larger M, the CRT satisfies the real-time signal processing time due to its parallel means of computation and there is a constant limit to this approach [8] The process of converting from conventional representations to RNS is known as forward conversion whilst converting from the RNS to the conventional representations is known as the reverse conversion.

The residue to conventional number representation is done mainly by the MRC or the CRT, [8]. The MRC is carried out by a weighted approach. The MRC is expressed by the following equations;

$$\mathrm{X} = a_1 + a_2 m_1 + a_3 m_1 m_2 + a_n m_1 m_2 m_3 \ldots m_{k-1} \qquad (8)$$

where $a_{i,i=1,k}$ is the Mixed Radix Digits (MRDs) can be computed as:

$$a_1 = x_1$$
$$a_2 = \left|(x_2 - a_1)|m_1^{-1}|_{m_2}\right|_{m_2}$$
$$a_3 = \left|((x_3 - a_1)|m_1^{-1}|_{m_3} - a_2)|m_2^{-1}|_{m_3}\right|_{m_3}$$
$$\vdots$$
$$a_k = \Big|(((x_k - a_1)|m_1^{-1}|_{m_k} - a_2)|m_2^{-1}|_{m_k} - \ldots - a_{k-1})|m_{k-1}^{-1}|_{m_k}\Big|_{m_k} \qquad (9)$$

This paper presents an efficient algorithm for detection and correction of single bit errors for the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1, 2^{2n}-3, 2^{2n}+1\}$

The rest of this paper is organized as follows: Section 4 presents the proposed method. In Section 5, the hardware implementation of the proposed scheme is presented, a simplified algorithm with numerical illustrations are also presented. The performance of the proposed scheme is evaluated in Section 6 whiles the paper is concluded in Section 7.

3. PROPOSED METHOD

This section provides a new method for detecting and correcting multiple bit errors in RRNS in the given moduli set.

3.1 Proposed Algorithm

The algorithm for the proposed scheme is given below;

1. Compute $\bar{y}$ from the received vector y using the MRC
2. Perform iterations using $C_t^n = \frac{n!}{(n-t)!t!}$ by discarding two residues at a time
3. If $\bar{y}$ falls within the legitimate range, stop and output $\bar{y}$. Declare there are no errors.
4. If $\bar{y}$ falls outside the legitimate range, calculate the residue vector y and the Hamming Distance $d(r_i, y)$. If $d(r_i, y) \leq t$, stop and output the result.
5. If $d(r_i, y) > t$, indicate that there are more than t errors and stop.

The process of recovering the original integer requires the received integer y be calculated using MRC from the set of received residues. From there, recovering the original integer only involves the modulo operation over several iterations. The algorithm is premised on the MRC and the HD.

For the given moduli set $S = \{2^n-1, 2^n, 2^n+1, 2^{n+1}-1, 2^{2n}-3, 2^{2n}+1\}$where $m_1 = 2^n-1, m_2 = 2^n, m_3 = 2^n+1, m_4 = 2^{n+1}-1, m_5 = 2^{2n}-3$ and $m_6 = 2^{2n}+1$.

The multiplicative inverses for the MRC based on the same moduli set are computed as follows;

$$|m_1^{-1}|m_2 = |(2^n-1)^{-1}|_{2^n} = -1 \qquad (10)$$

$$|m_2^{-1}|m_3 = |(2^n)^{-1}|_{2^n+1} = 1 \qquad (11)$$

$$|m_3^{-1}|m_4 = |(2^n+1)^{-1}|_{2^{n+1}-1} = 2^n \qquad (12)$$

$$|m_3^{-1}|m_5 = |(2^n+1)^{-1}|_{2^{2n}-3} = 2^n \qquad (13)$$

$$|m_4^{-1}|m_5 = |(2^{n+1}-1)^{-1}|_{2^{2n}-3} = 2^{n+1}-6 \qquad (14)$$

$$|m_2^{-1}|m_4 = |(2^n)^{-1}|_{2^{n+1}-1} = 2^n-1 \qquad (15)$$

$$|m_3^{-1}|m_6 = |(2^n+1)^{-1}|_{2^{2n}+1} = 2^n+3 \qquad (16)$$

$$|m_4^{-1}|m_6 = |(2^{n+1}-1)^{-1}|_{2^{2n}+1} = 2^{n+1}-3 \qquad (17)$$

$$\left|m_5^{-1}\right|m_6 = |(2^{2n}-3)^{-1}|_{2^{2n}+1} = 2^{2n}-12 \qquad (18)$$

Now the $a_{i's}$ can be computed using the MRC as follows;

$$a_1{=}x_1 \qquad (19)$$

$$a_2{=} \quad | \quad (x_2 - x_1)|m_1^{-1}|m_2|m_2 = |\,(x_2 - x_1)(-1)|m_2 = |x_1 - x_2|_{2^n} \qquad (20)$$

The decimal equivalent for the non-redundant part is thus;

$$X = a_1 + a_2 m_1 \qquad (21)$$

$$= a_1 + a_2(2^n-1) \qquad (22)$$

The redundant part is;

$$a_3{=}x_3 \qquad (23)$$

$$a_4{=} \;|\; (x_4 - x_3)|m_3^{-1}|m_4|m_4 = |\,(x_4 - x_3)(2^n+1)|m_2 = |2^n x_4 - 2^n x_3 + x_4 - x_3|_{2^{n+1}-1} \qquad (24)$$

$$a_5{=} \;|\; (x_5 - x_3)|m_3^{-1}|m_5 \quad -a_4) \quad |m_4^{-1}|m_5|m_5 = |(x_5 - a_3)(2^n) - (2^n x_4 - 2^n x_3 + x_4 - x_3)(2^{n+1} - 6|_{2^{2n}-3} \qquad (25)$$

$$a_6{=}\,|\,(x_6 - a_3)|m_3^{-1}|m_6 \text{ -}a_4)\ |m_4^{-1}|m_6\text{- } a_5)|m_5^{-1}|m_6|m_6$$

$$= |((x_6 - a_3)(2^n+3)) - (2^n x_4 - 2^n x_3 + x_4 - x_3)(2^{n+1}-3) - a_5)(2^{n+1}-3))|_{2^{2n}+1} \qquad (26)$$

The decimal equivalent for the redundant part is thus;

$$\bar{X} = a_3 + a_4 m_1 + a_5 m_1 m_2 + a_6 m_1 m_2 m_3 \qquad (27)$$

$$= a_3 + a_4(2^n-1) + a_5(2^n-1)(2^n) + a_6(2^n-1)(2^n)(2^n+1) \qquad (28)$$

3.2 Hardware Implementation

For the considered moduli set for the non-redundant part $S = \{2^n-1, 2^n\}$ with its corresponding residues (x_1, x_2). We let the binary representations of the residues be;

$$x_1 = x_{1,n-1} \dots x_{1,1} x_{1,0} \qquad (29)$$

$$x_2 = x_{2,n-1} \dots x_{2,1} x_{2,0} \qquad (30)$$

and the redundant part as;

$$x_3 = x_{3,n,} x_{3,n-1,} x_{3,n-2,} \dots x_{3,0} \qquad (31)$$

$$x_4 = x_{4,n+1,n-1,n-2} \dots x_{4,0} \qquad (32)$$

$$x_5 = x_{5,2n,} x_{5,2n-1,} x_{5,2n-2,} \dots x_{5,0} \qquad (33)$$

$$x_6 = x_{6,2n,} x_{6,2n-1,} x_{6,2n-2,} \dots x_{6,0} \qquad (34)$$

Thus by the MRC technique,

$$a_1{=}\,x_1 \qquad (35)$$

$$a_2 = |x_1 + \bar{x}_2|_{2^n}$$

$$= \left|x_{1,n-1} x_{1,n-2} \dots x_{1,1} x_{1,0} + \bar{x}_{2,n-1} \bar{x}_{2,n-2} \dots \bar{x}_{2,1}\, \bar{x}_{2,0}\right|_{2^n}$$

$$= \left|\underbrace{x_{1,n-1} x_{1,n-2} \dots x_{1,1} x_{1,0}}_{\text{n bits}} + \underbrace{\bar{x}_{2,n-1} \bar{x}_{2,n-2} \dots \bar{x}_{2,1}\, \bar{x}_{2,0}}_{\text{n bits}}\right|_{2^n} \qquad (36)$$

Implementation of equations (19) – (36) gives the correct output of a_2 whenever an error occurs in the non-redundant part. The decimal representation of the architecture for the non-redundant from equations (21) and (22) is;

$$X = a_1 + a_2 m_1 \qquad (37)$$

$$= a_1 + a_2(2^n-1) \ (38)$$

Also, the decimal representation of the redundant part is thus;

$$a_3{=}x_3 \qquad (39)$$

$$a_4{=} \;|\; (x_4 - x_3)|m_3^{-1}|m_4|m_4 = |\,(x_4 - x_3)(2^n+1)|m_2 = |2^n x_4 - 2^n x_3 + x_4 - x_3|_{2^{n+1}-1} \qquad (40)$$

$$a_5{=} \;|\; (x_5 - x_3)|m_3^{-1}|m_5 \quad -a_4) \quad |m_4^{-1}|m_5|m_5 = |(x_5 - a_3)(2^n) - (2^n x_4 - 2^n x_3 + x_4 - x_3)(2^{n+1} - 6|_{2^{2n}-3} \ (41)$$

$$a_6{=}\,|\,(x_6 - a_3)|m_3^{-1}|m_6 \text{ -}a_4)\ |m_4^{-1}|m_6\text{- } a_5)|m_5^{-1}|m_6|m_6$$

$= |((x_6 - a_3)(2^n + 3)) - (2^n x_4 - 2^n x_3 + x_4 - x_3)(2^{n+1} - 3) - a_5)(2^{n+1} - 3))|_{2^{2n}+1}$ (42)

The decimal equivalent for the redundant part is thus;

$X = a_3 + a_4 m_1 + a_5 m_1 m_2 + a_6 m_1 m_2 m_3$

$= a_3 + a_4(2^n - 1) + a_5(2^n - 1)(2^n) + a_6(2^n - 1)(2^n)(2^n + 1)$ (43)

3.3 Proposed Architecture

The proposed architecture used for the detection and correction of multiple bit errors is based on the MRC. The Mixed Radix Digits (MRDs) are computed according to equation (21) where all the MRDs a_1 and a_2 are individually computed in collaboration with its respective modulus which is seen from equations (19) and (20). The a_{is} which represent the non-redundant part are computed separately which is seen in Figure (1). As shown in Figure (1), a_2 is computed using one regular CPA which has a delay of $2n\Delta_{FA}$ and an area of $n\Delta_{FA}$ each. In order to obtain the MRD, a_2 will require a total area of $n\Delta_{FA}$. The total delay required for the proposed scheme is $2nD_{FA}$. The schematic diagram for the proposed scheme is shown in Figure (1).

The schematic diagram for the proposed scheme is shown below:

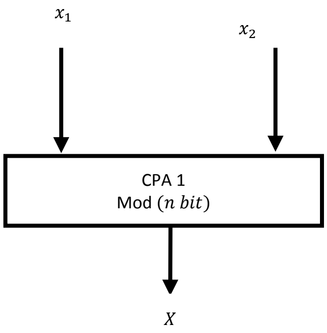

Figure 1: Block Diagram of RC for the non-redundant part

3.4 Numerical Results

Let us now consider some numerical illustrations with the proposed scheme.

Consider an (n, k) code where n is the length of the code and k is the dimension of the code with the moduli set $m_{1,} m_2, m_3, m_4, m_5 = (3,4,5,7,13,17)$ where m_1 and m_2 are non-redundant moduli, $m_{3,}$ $m_{4,} m_{5,}$ and m_6 are the redundant moduli. We consider the integer message y=11, for its residue digits are $y_i = (2, 3, 1, 4, 11, 17)$. The legitimate range = M_R= 3*4=12 and the illegitimate range = M_K= 5*7*13*17 =7735. Assume that during storage or computation, an error occurs in the second and sixth residues respectively, i.e. two errors (t=2) have propagated into y during transmission using $U_1 = 2$ and $U_2 = 6$ as error positions. Therefore, the received codevector will be $\bar{y}_i = (2, \bar{5}, 1, 4, 11, , \bar{7})$.

The decoding process gives;

$y_1, r_2, r_3, r_4 - X_{1234} = 221$

$r_1, r_2, r_3, r_5 - X_{1235} = 401$

$r_1, r_2, r_4, r_5 - X_{1245} = 557$

$r_1, r_3, r_4, r_5 - X_{1345} = 11 *$

$r_2, r_3, r_4, r_5 - X_{2345} = 1441$

$r_1, r_2, r_3, r_6 - X_{1236} = 41$

$r_1, r_2, r_4, r_6 - X_{1246} = 1061$

$r_1, r_3, r_4, r_6 - X_{1346} = 275$

$r_2, r_3, r_4, r_6 - X_{2346} = 1061$

$r_1, r_2, r_5, r_6 - X_{1256} = 245$

$r_1, r_3, r_5, r_6 - X_{1356} = 1259$

$r_2, r_3, r_5, r_6 - X_{2356} = 3781$

$r_1, r_4, r_5, r_6 - X_{1456} = 3560$

$r_2, r_4, r_5, r_6 - X_{2456} = 2013$

$r_3, r_4, r_5, r_6 - X_{3456} = 4886$

From these results, we can observe that whenever r_2 and r_6 are involved in the computation they give an illegitimate value and also falls within the illegitimate range i.e. X_{1234}, X_{1235}, X_{1245}, X_{2345}, X_{1236}, X_{1246}, X_{1346}, X_{2346}, X_{1256}, X_{1356}, X_{2356}, X_{1456}, X_{2456} and X_{3456}. When r_2 and r_6 were discarded in the X_{1345}, the recovered data is 11 and also falls within the legitimate range, which clearly indicates that r_2 and r_6 are the erroneous digits. We can conclude that, the correct result is 11 and there were errors in r_2 and r_6 which can be corrected by computing $r_2 = 11 \bmod 4 = 3$ and $r_6 = 11 \bmod 17 = 11$.

Table 1: The Residue Vectors and Hamming Distances for Residue Digits Error Correction

i	$\bar{y}$	r_i	y	$d(r_i, y)$
1	221	2,1,1,4,0,0	2,3,1,4,11,11	3
2	401	2,1,1,2,11,10	2,3,1,4,11,11	3
3	557	2,1,2,4,11,13	2,3,1,4,11,11	3
4	11	2,3,1,4,11,11	2,3,1,4,11,11	0*
5	1441	1,1,1,6,11,13	2,3,1,4,11,11	3
6	41	2,1,1,6,2,7	2,3,1,4,11,11	4
7	1061	2,1,1,4,8,7	2,3,1,4,11,11	3
8	275	2,3,0,2,2,3	2,3,1,4,11,11	4
9	1061	2,1,1,4,8,7	2,3,1,4,11,11	3
10	245	2,1,0,0,11,7	2,3,1,4,11,11	4

Table 1 shows the HD between any two codevectors that are computed based on theorem 5 of the HD theorems from the paper. From the comparison involving the residue vectors and the Hamming Distances, the only value that falls within the legitimate range and has a Hamming Distance $d(r_i, y)$ which is less than or equal to 2 i.e $(t \leq 2)$ is 11. This indicates that, the algorithm has correctly detected and corrected the transmitted integer message. The proposed algorithm is simple and allows the original integer to be recovered without having to use large integers.

4. PERFORMANCE EVALUATION

In evaluating the performance of the proposed algorithm, the paper is compared with similar best known state-of-the-art scheme. The theoretical analysis performed between the proposed scheme and the scheme presented by Amusa &

Nwoye [10] showed that the proposed scheme has an area of n whereas the scheme proposed by Amusa & Nwoye [10] is $5n + 2$. It was observed that the processing speed for the architecture presented by Amusa & Nwoye [10] is $6n + 2$ and the proposed scheme has a processing speed of $2n$. On the area and delay analysis, both schemes employed simple adders for the design of the architecture. Comparatively, the proposed scheme presents simpler architecture which requires less hardware resources and has a better processing speed and computing time than the scheme presented by Amusa & Nwoye [10] for different values of n. When n becomes large in both schemes, the proposed scheme tends to be simpler and has less delay than that of Amusa & Nwoye [10] as shown in Figures 2 and 3.

Table 2: Area, Delay Comparison

Scheme	Area(Δ_{FA})	Delay(D_{FA})
[10]	$5n + 2$	$6n + 2$
Proposed	n	$2n$

Figures 2 and 3 show the graphical illustrations of the area and delay comparisons.

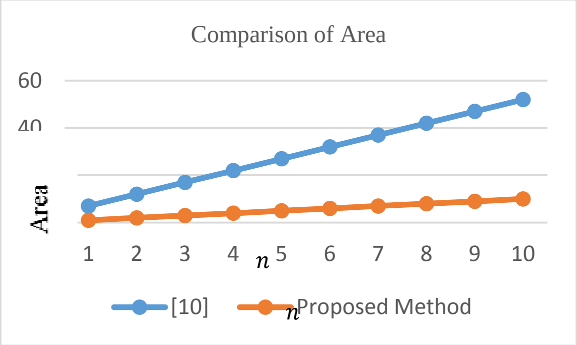

Figure 2: Graph of area comparison of proposed scheme with Amusa & Nwoye [10]

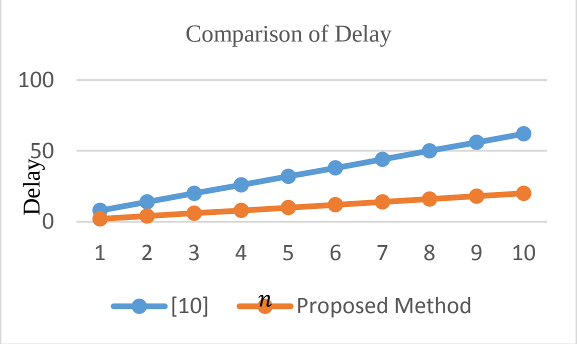

Figure 3: Graph of delay comparison of proposed scheme with Amusa & Nwoye [10]

5. CONCLUSION

The number of redundant moduli governs the numbers of detectable and correctable residue digit errors in RNS. RRNS is capable of detecting and correcting residue errors as a result of redundancy. It is an established fact that decoding using the CRT requires large modulo operation. The legitimate range represents the useful computational residues that are error-free while the illegitimate range is useful for error detection. The proposed algorithm is premised on the MRC and the HD as a joint technique to detect and correct multiple bits errors. The simplicity of the proposed algorithm can easily locate errors in a channel. The proposed algorithm and that of Amusa & Nwoye [10] both employed simple adders in the architectural design. However, the proposed scheme incorporated the HD that simplified the hardware design and this resulted in improvement of the speed by 68% and tends to require about 81% less hardware resources.

6. ACKNOWLEDGMENTS

The authors would like to express their gratitude to the Editor and reviewers for their thorough examination of the manuscript and the invaluable feedback.

7. REFERENCES

[1] W. Wei, M.N.S. Swamy, M.O. Ahmad, ”RNS application for digital image processing”, *Proceedings of the 4th IEEE international workshop on system-on-chip for real time applications, Canada*, (2004), pp. 77-80.

[2] M.I. Daabo, & K.A. Gbolagade (2014). An Overflow Detection Scheme with a Reverse Converter for the Moduli set $\{2^n - 1, 2^n, 2^n + 1\}$. *Journal of Emerging Trends in Computing and Information Sciences*, ISSN 2079-8407, Vol. 5, No. 12, pp. 931-935.

[3] S. Yen, S. Kim, S. Lim, S. Moon, ”RSA speedup with Chinese remainder theorem immune against hardware fault cryptanalysis”, *IEEE Transactions on Computers*, Vol. 52, No. 4, (2003), pp. 461-472.

[4] E. Kinoshita & K. Lee, ”A residue arithmetic extension for reliable scientific computation”, *IEEE Transactions on Computers*, Vol. 46, No. 2, (1997), pp. 129-138.

[5] R. Convey & J. Nelson, ”Improved RNS FIR filter architectures”, *IEEE Transactions on Circuits and Systems-II*, Vol. 51, No. 1, (2004), pp. 26-28.

[6] H. Krishna, K. Lin, & J. Sun, (1992). A coding theory approach to error control in redundant residue number systems—Part I: Theory and single error correction,” *IEEE Trans. Circuits Syst.*, Vol. 39, No. 1, pp. 8–17.

[7] J. Sun & H. Krishna, (1992). A coding theory approach to error control in redundant residue number systems—Part II: Multiple error detection and correction. *IEEE Trans. Circuits Syst.*, Vol. 39, No. 1, pp. 18–34.

[8] L. Yang & L. Hanzo, *Coding theory and performance of redundant residue number system codes*. [Online]. Available: http://www-mobile. ecs.soton.ac.uk/

[9] V.T. Goh & M. Siddiqi, (2008). Multiple error detection and correction based on redundant residue number systems, *IEEE Transactions on Communications*, Vol. 56, No. 3, pp. 325–330.

[10] K.A. Amusa, & E.O. Nwoye, (2012). Novel Algorithm for Decoding Redundant Residue Number Systems (RRNS) Codes. *International Journal of Research and Revies in Applied Sciences(IJRRAS)*, Vol. 12, No. 1, pp. 158-163.

[11] T.F. Tay, & Chip-Hong C. (2016). A Non-Iterative Multiple Residue Digit Error Detection and Correction

Algorithm in RRNS. *IEEE transactions on computers,* Vol. 65, No. 2.

[12] C. Ding, D. Pei, & A. Salomma, (1996). Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography. Singapore: World Scientific Publishing.

[13] J.H. McClellan, J.H. & C.M. Rader (1979). Number theory in Digital Signal Processing. Englewood Cliffs, N.J: Prentice Hall.

[14] W.W. Peterson & E. Weldon Jnr, (1972). Error Correcting Codes. Cambridge: MA: MIT Press.