Journal of Advances in Mathematics and Computer Science

36(8): 27-43, 2021; Article no.JAMCS.72725
ISSN: 2456-9968
(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

A New Approach to Detecting and Correcting Single and Multiple Errors in Wireless Sensor Networks

Yakubu Abdul-Wahab Nawusu[1*], Alhassan Abdul-Barik[2] and Salifu Abdul-Mumin[3]

[1]Computer Science Department, Tamale Technical University, Tamale, Ghana.
[2]Computer Science Department, University for Development Studies, Tamale, Ghana.
[3]Information Systems and Technology Department, C.K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana.

Authors' contributions

This work was carried out in collaboration among all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JAMCS/2021/v36i830388
Editor(s):
(1) Dr. Dariusz Jacek Jakóbczak, Koszalin University of Technology, Poland.
Reviewers:
(1) Gopal Chand Gautam, India.
(2) Manish Bhardwaj, India.
Complete Peer review History: https://www.sdiarticle4.com/review-history/72725

Original Research Article

Received 29 June 2021
Accepted 09 September 2021
Published 22 September 2021

Abstract

Transmission errors are commonplace in communication systems. Wireless sensor networks like many other communication systems are susceptible to various forms of errors arising from sheer noise, heat and interference in sensor circuitry and from other forms of distortions. Research efforts in WSN have attempted to guarantee reliable and accurate data transmission from a target environment in the midst of these unwanted exposures. Many techniques have appeared and employed over the years to deal with the issue of transmission errors in communication systems. In this paper we present a new approach for single and multiple error control in WSN relying on the inherent fault tolerant feature of the Redundant Residue Number System. As an off shoot of Residue Number System, RRNS's fault tolerant capabilities help in building robust systems required for reliable data transmission in WSN systems. The Chinese Remainder Theorem and the Manhattan Distance Heuristics are used during the integer recovery process when detecting and correcting error digit(s) in a transmitted data. The proposed method performs considerably better in terms of data retrieval time than similar approaches by needing a smaller number of iterations to recover an originally

**Corresponding author: Email: nabdul-wahab@tatu.edu.gh;*

transmitted data from its erroneous form. The approach in this work is also less computationally intensive compared to recent techniques during the error correction steps. Evidence of utility of the technique is illustrated in numerical examples.

Keywords: Residue number system; error control; wireless sensor network; transmission error.

1 Introduction

Wireless sensor network (WSN) is an ensemble of many tiny intelligent sensing agents that run on low capacity batteries. WSN has gained widespread application especially for remote data sensing and transmission in both accessible and inaccessible environments. WSN has thus far been deployed in areas such as environmental monitoring, health-care, transportation, structural health monitoring, water and air quality monitoring and many other areas. Many of these environments are hash causing either a rapid depletion of the already limited energy resource or degenerating the quality of the transmitted data especially when a robust and efficient communication protocol is absent. At worse, wireless links in WSN suffer from packet loses, making reliability of packet delivery difficult to achieve. Moreover, because the transmission strength of sensor nodes is limited, sensed data are often delivered to target user bases via lossy paths, further diminishing the reliability of the data delivery. Many conventional techniques have been applied to improve transmission reliability in communication systems including in WSN to varying degrees of success. Transmission control protocol and retransmission/acknowledgment (ACK) are traditionally used in the transport layer and media access control (MAC) layer respectively. Reed-Solomon (RS) code offer reduce transmission overhead by allowing a receiver to decode the original data using a received fixed number of encoded data.

Opportunistic coding, ruby transform and tornado code have been applied as coding based reliable packet delivery mechanisms. The specific hurdle with building reliable data transmission schemes for WSN is the limitation in the battery capacity of sensor nodes. Replacing these batteries in some deployment areas is difficult or dangerous or even wholly impossible to do. A fine balance is required therefore for two of the important design metrics in wireless communication systems; link reliability and reduced power dissipation. A meaningful reliability scheme needs to pay attention to the longevity of the battery base of the sensor nodes as well as the reliability of data delivery. Exiting schemes that rely on capping node's transmission power as a node saving technique suffer from unreliable links many times. Similarly, schemes that enforce step-wise node activity suffer from packet drops and retransmissions. The fact that several retransmission requests might be made of an erroneous message makes it unsuited to the efficient energy needs for WSN. It is opined by [1] that, the cost of transmitting a bit of information is equal to the cost of processing thousand sensor related jobs. Consequently, the need for possible several retransmissions causes enormous amount of energy dissipation in sensor nodes making it not ideal to control errors in WSN. The need arises therefore to generate an energy efficient and reliable packet delivery scheme for WSN. The coding method used here is based on RRNS designed to control erroneous transmission and offer reliability in data transmission with specific application interest in WSN. Redundant RNS provides favourable reliability requirements due to its streamlined parallel processing ability. Additionally, it’s borrow-free and carry-free features benefits error control efforts since errors present in one modulus are not propagated to others [2]. These favourable capabilities in the RRNS have been utilized in electronic and communications systems to improve performance and reliability [3] in application areas such as wireless local area network, space-time block codes, multicarrier modulation, cloud storage services and wireless sensor network [4]. The research interest in this work is thus to offer a reliable approach to deal with and recover to original forms errors that creep into transmitted data in wireless networks in general and specially in wireless sensor networks.

2 Review of Related Literature

Error detection and control techniques are an important means for reliable signal transmission in communication systems. Error detection and error correction are therefore important tasks that are required to be performed when communication takes place between network nodes. Error correction in digital system such as WSN are enabled by a number of methods including the Redundant Residue Number System; a number system that extends the RNS. Authors in [5] employed the properties of RNS and noticed its inherent benefits for high speed operations in computing applications. The added advantage when an adequate amount of redundancy is added is that, it is able to correct arithmetic processing errors from many sources including transmission and storage noise [6]. The telling reason is that, the carry feature is not present in RNS; thus, errors in one residue position are confined and are not propagated from one residue to the other offering great speed as a result. Additionally, residue positions are insignificant with respect to each other, allowing an integer value to be computed from remaining residues when some redundant residues are lost.

The first proposal for error correction using RRNS appears in [7]. The method corrects single-bit error in binary-coded residues. The error correction approach in [7] however, is lengthy and offers no real multiple error correction benefits beyond a single error. A rather complex method is presented in [5] that is capable of single residue error correction. The breakthrough in self-checking computations based on residue arithmetic appeared however in [8]. The authors presented a table-based method for correcting single residue-error. Restrictive sufficient and necessary conditions constraining moduli selection for error correction are required coupled with the huge memory requirements of the correction table. Their method thus becomes impractical for multiple error correction. The implementation of a table-based error correction was eliminated in [9] using approximate computations. Earlier to the work of [9], the work in [10] demonstrated that it is possible to spot and correct a single residue-error based on the basic rule of using two redundant moduli. Mandelbaum op. cit., posited that a RRNS with r redundant moduli will detect and correct $\lfloor r/_2 \rfloor$ errors, where $\lfloor \emptyset \rfloor$ is the largest integer such that $\lfloor \emptyset \rfloor \leq \emptyset$. The codes in [10] utilized the Chinese Remainder Theorem and were considered only efficient for single-error correction and not applicable to multiple errors control. Multiple errors correction technique is presented in an extended work by same authors, by means of the continued fractions and Euclid's algorithms premised on some sufficiently provided conditions. The work in [11] brought to the limelight the error correcting properties of RRNS and popularized the concepts of legitimate and illegitimate range. They presented proof to conditions necessary and sufficient for correcting errors in a legitimate number. They also offered a procedure to detect and correct errors upon the determination of whether a number is legitimate or illegitimate [3,12] and based upon the values of projections derived from redundant RNS. They further showed that an introduction of a single digit error into a legitimate number, X will transform it (the number) into another number in the illegitimate range of the RRNS.

Single error correction procedures are presented in [13] using modulo projections. In [14] an algorithm is presented for scaling and detecting and correcting single digit error by building a lookup table with redundant digits of MRC. Error detection and correction using moduli set with common factors is presented in [15]. Methods that are based on RRNS requires forward conversion and backward conversion. These are crucial to the error detection and correction schemes that are based on RRNS. Mixed radix conversion (MRC) and Base extension (BEX) have been used with RRNS in checking residue number errors in digital signal processing application [3] arising from efficient pipelining architectures. The authors in [16] employed Chinese Remainder Theorem (CRT) to decode received residues into their integer equivalent. They extended their work to control multiple errors in RRNS codes in a later research. In both researches, no architecture was provided for forward and reverse conversion. The modulo operation involving a large M makes the approach in [16] computationally intensive. An architecture for a single residue digit error correction and multiple residue digits' error correction appear in [17]. [18] proposes a method for decoding RRNS codes using the matrix method proposed in [19] and modulo projection. The scheme in [18] is applicable to multiple error detection and correction but the procedures use of the mixed radix method may slow the encoding process. In recent past, RRNS has been applied to achieve fault tolerance in communication systems for reliable transmission; data encryption and compression; and in cryptographic and stenographic schemes [20,21]. The high degree of computational parallelism and carry-free operations inherent in the system offers new avenue for energy efficiency, for example in sensor nodes, data security, increased data transfer rate, and better data storage. In [22], some insight is given into the applicability of RNS in WSN. The authors claim to have achieved in their studies reduced traffic rate in wireless sensor network with decrease amount of data transmission and by extension, a reduction in power consumption of sensor nodes. The work in [22] also offered error detection and correction steps for single errors. However, multiple error correction was not confirmed in their work. [23] suggested a novel method for single error detection and correction in digital communication systems using MRC and single consistency check. [24] applied RNS for energy efficient channel coding of physiological signals in Wireless Body Area Networks (WBAN) with the design goal of achieving low power consumption through the avoidance of retransmissions. The authors presented a reverse converter for a $5n$ bit moduli set $\{2^n + 3, 2^n + 1, 2^n, 2^n - 1, 2^n - 3\}$ based on the Mixed Radix conversion technique. In addition to the low energy consumption gains, the use of RNS according to the authors allowed bust errors in WBAN to be handled even though no algorithm was presented in that regard. [25] presented a highly secured data encryption and decryption scheme for enhancing the traditional Huffman's method using RNS. The scheme in [25] uses 6 channel moduli set made up of four information moduli; $\{2^{n-1}, 2^n - 1, 2^n + 1, 2^{n+1} - 1\}$ and two redundant moduli; $\{2^{2n} - 3,\ 2^{2n} + 1\}$ for purposes of error control. The scheme was demonstrably cost efficient in terms of data transmission and storage. It (the scheme) however, is not applicable for odd values of n, which generated a non-relatively prime set of moduli. [26] presents a novel LZW-RNS compression scheme based on the traditional moduli set $\{2^n - 1, 2^n, 2^n + 1\}$. The scheme therein performed better compared to the traditional LZW algorithm in aspects of compression efficiency, security, and execution time. The scope of the work did not include error detection and correction. An extension of the work in [26] appears in [27] that supports single error correction in encrypted and compressed data. An additional layer of security is enforced in their method by constraining the encoding and decoding processes to only work for even values of n. [28] extends the work of [23] with a multiple error detection, location and correction algorithm using MRC and double consistency check.

Salifu and Gbolagade [29] offered a single error detection and correction scheme based on the redundant residue number system using the 6n dynamic range moduli set $\{2^{2n} + 1,\ 2^{n+1} + 1,\ 2^{n+1} - 1,\ 2^n + 1,\ 2^n\}$. Since communications systems are exposed to more than just single error, their scheme is limited in detecting and correcting multiple errors. The complexity of their architecture limits it performance gains because of the presence of the two moduli, $\{2^{n+1} + 1,\ 2^{n+1}\}$. [30] applied RNS to the Lempel-Ziv-Welch data compression algorithm. The authors touted the efficiency of their scheme than the traditional LZW compression in terms of security, compression efficiency, speed of execution, and fault tolerance. The research proposed a forward conversion architecture based on the moduli set, $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1, 2^{2n} - 3\}$. The presence of two 2^{k+1} type moduli potentially increase the hardware complexity of their error recovery process. Additionally, a reverse converter was not generated. Also, the error detection and correction scheme they proposed applied only to single error. Recently, [31] extended the work in [30] and [29] by generating error detection and handling procedure that allowed both single and multiple residue errors to be detected and corrected. The procedure in [31] was designed for application in stenographic and cryptographic systems, nonetheless. A reverse converter base on the traditional CRT was also presented unlike in both [30] and [29] where no reverse conversion architectures were suggested. The downside to the scheme in [31] is that, the integer recovering process requires a substantial number of iterations of at most C_t^n, consequently reducing the data retrieval speed. This work herein presents an improvement in the error detection and correction research in communication systems with emphasis on its application in WSN. The goal particularly is to offer a mechanism that minimizes the data recovery time.

2.1 Background of residue number system

Residue Number Systems (RNS) has its roots in the ancient book of Sun Tzu [31]. Its revival however started in the 1950s, as an alternative number system for applications requiring fast arithmetic and fault-tolerant operations [22]. The RNS encodes a number as a set of its remainders with respect to a specified set of relatively prime moduli. Many of its innate features make it useful and attractive for special-purpose computations. For example, RNS provides no carry mechanism, allowing addition and multiplication to be done in parallel with no interaction between digits. Also, the allowance of parallel and carry-free computations offers the benefit of fast arithmetic operations when compared with straight binary encoding [5]. Furthermore, because residues reveal no weight information, error in any of the residue positions is not propagated to other digit positions.

RNS also offers some useful properties for error detection and correction in computerized systems. These and many of its inherent features helps in building fault tolerant systems required of many data communications systems including in digital filtering, convolution, Discrete Cosine Transform, communication engineering, cryptography, image processing and speech processing and stenography schemes [32,33,21,22]. Others beneficiary areas of application of RNS are; direct digital frequency synthesis, Discrete Fourier Transform and Fast Fourier Transform [3]. Some bottlenecks however still exist that limits the general implementation of RNS especially in general purpose computing. A residue number system is defined by a set of pair-wise relatively prime integers called the moduli set. The moduli set is denoted as $\{m_1, m_2, \ldots, m_n\}$, $i = 1, 2, \ldots, n$ that $GCD\ (m_i, m_j) = 1$ for $i \neq j$, where $GCD\ (m_i, m_j)$ means the greatest common divisor of m_i and m_j. A given RNS is capable of uniquely representing all integers that lie in its dynamic range, M [34] given in Equation 1. Given the moduli set $\{m_1, m_2, \ldots, m_n\}$, the dynamic range for positive integers is denoted in Equation (1) and corresponds to a range of all positive integers from 0 to $M - 1$.

$$M = \prod_{i=1}^{n} m_i \tag{1}$$

Encoding a decimal number into an RNS code (called forward conversion) is often not a challenge. However, many applications based on RNS require data in residue representation to be converted to a weighted number system - binary or decimal - in order for the encoded data to be used. The typical structure of an RNS processor that supports conversion from conversional number system (CNS) - binary or decimal - to an RNS number is shown in Fig. 1 below.

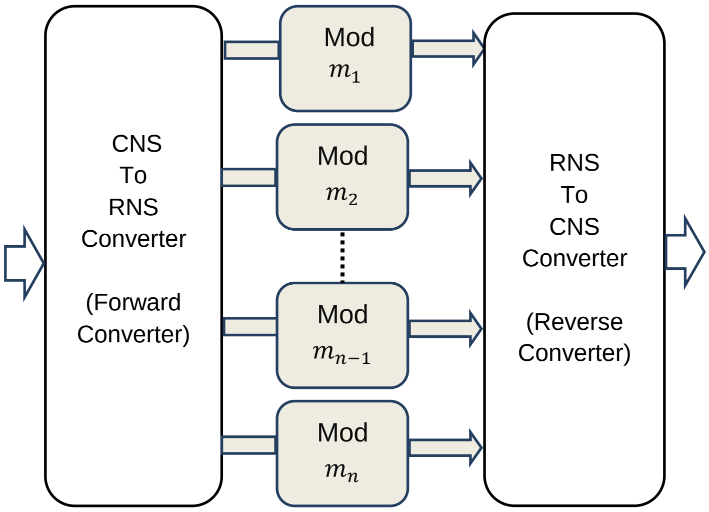

Fig. 1. Structure of an RNS processor

There are many techniques used to achieve RNS to weighted number systems conversion. The popular ones are the Chinese Remainder Theorem and the Mixed Radix Conversion [15,35,34]. The Chinese Remainder Theorem (CRT) is defined as follow; Given a moduli set $\{m_1, m_2, \dots, m_n\}$, an equivalent decimal integer X can be derived from its residues $\{x_1, x_2, \dots, x_n\}$ using the CRT as follows:

$$X = \left| \sum_{i=1}^{n} M_i |x_i M_i^{-1}|_{m_i} \right|_M \tag{2}$$

Where, M is given as in Equation (1); $M_i = {}^{M}/_{m_i}$; $|M_i^{-1} M_i|_{m_i} = 1$.

2.2 Background of redundant residue number system

Redundant Residue Number System (RRNS) is an extension of RNS which is engineered by having extra residues added to an original information residue set. Because of this, RRNS inherent the capabilities of RNS especially in relation to fault tolerance in addition to the provision of error detection and correction benefits. In both RNS and RRNS, the Chinese Reminder Theorem is commonly applied to recover received data bits so as to determine the presence or otherwise of error(s). The foundation of RRNS starts with a selection of a set of n pairwise relatively prime positive integers $\{m_1, m_2, \dots, m_k, \dots, m_{k+r}\}$, chosen such that, the greatest common divisor, $GCD\left(m_i, m_j\right) = 1$ for each pair of i and j such that $i \neq j$, and $m_1 < m_2 < \cdots < m_k < m_{k+1} < \cdots < m_n$. From this set of n moduli, the first k moduli form a set of non-redundant moduli while the remaining $r = n - k$ moduli form the set of redundant moduli. The redundant parts are used for error detection and correction [16]. The residue digits $\{x_1, x_2, \dots, x_k\}$ are the redundant residue digits whereas $\{x_{k+1}, x_{k+2}, \dots, x_{k+r}\}$ are the redundant residue digits. An RRNS number therefore is represented by a total of $r + k$ residue digits. The product of the non-redundant moduli set form the legitimate range, $[0,\ M_k)$ and the product of the remaining $n - k$ redundant moduli form the illegitimate range, $[M_K,\ M_R)$ Both M_K and M_R can be computed using Equation (1). The total range, $[0; M_T)$, defines the set of states represented by the RRNS and is given as;

$$M_T = M_K \times M_R \quad (3)$$

Given any integer X in the range of $[0; M)$, where M is as in Equation (1), X can be uniquely represented as a residue vector $x = \{x_1, x_2, \dots, x_n\}$. Each of the residues x_i corresponds to X modulo m_i such that $0 \leq x_i < m_i$. However, for error correction to work, X has to be selected from the range of $[0, MK)$ instead, where M_K is computed from Equation (1) [25,36]. Thus, the residue vector X can be divided into two parts, namely the first k residues called information residues and the remaining r residues called redundant residues [17]. Given a residue vector $\{x_i\}_{i=1,n}$, the corresponding integer X can be uniquely determined by simultaneously solving all n linear congruences [11]. The problem of simultaneously solving a set of n linear congruences is simplified by using Equation (2).

3 Methodology

3.1 Proposed forward converter for $\{\mathbf{2^n - 1},\ \mathbf{2^n}, \mathbf{2^{n+1} - 1},\ \mathbf{2^{2n} - 3},\ \mathbf{2^{2n} + 1},\ \}$

At the outset, the proposed RNS-based transmission scheme for WSN requires a sensed attribute value (for simplicity sake we will use an integer value for demonstration) to be encoded in RNS representation with respect to a given moduli set. This process is referred to as forward conversion. With regard to the moduli set in used, the first three represents the information moduli and the remaining two (referred to as the redundant moduli) is added for purposes of error detection and correction.

Given the moduli set $\{2^n - 1,\ 2^n, 2^{n+1} - 1,\ 2^{2n} - 3,\ 2^{2n} + 1\}$, $\forall\, n > 1$ and even, an integer X of width $3n + 1$, can be represented in binary as follows;

$$\text{X} \rightarrow \underbrace{X_{3n} \dots X_{2n}}_{B_{3,}\, n+1\ bits} | \underbrace{X_{2n-1} \dots X_n}_{B_{2,} n\ bits} | \underbrace{X_{n-1} \dots X_0}_{B_{1,} n\ bits} \quad (4)$$

Where, B_1, B_2 and B_3 are binary numbers given as:

$$B_1 = \sum_{i=0}^{n-1} x_i 2^i,$$

$$B_1 = \sum_{i=n}^{2n-1} x_i 2^{i-n} \quad (5)$$

$$B_1 = \sum_{i=2n}^{3n} x_i 2^{i-2n}.$$

Thus, X can be computed as;

$$X = B_1 + 2^n B_2 + 2^{2n} B_3 \quad (6)$$

The residues of X w.r.t to the moduli set; $\{2^n - 1,\ 2^n, 2^{n+1} - 1,\ 2^{2n} - 3,\ 2^{2n} + 1,\ \}$, can be derived as follows:

$$x_1 = |X|_{m_1}$$

$$= ||B_1|_{2^n-1} + |2^n B_2|_{2^n-1} + |2^{2n} B_2|_{2^n-1}|_{2^n-1}$$

$$= |B_1 + B_2 + B_3|_{2^n-1} \quad (7)$$

$$x_2 = |X|_{2^n},$$
$$= ||B_1|_{2^n} + |2^n B_2|_{2^n} + |2^{2n} B_3|_{2^n}|_{2^n},$$
$$= |B_1|_{2^n},$$
$$= B_1. \quad (8)$$

$$x_3 = |X|_{2^{n+1}-1}$$
$$= ||B_1|_{2^{n+1}-1} + |2^n B_2|_{2^{n+1}-1} + |2^{2n} B_3|_{2^{n+1}-1}|_{2^{n+1}-1}$$
$$= |B_1 + 2^n B_2 + 2^{n-1} B_3|_{2^{n+1}-1}$$
$$= |B + 2^{n-1} B_3|_{2^{n+1}-1}$$
$$= |B + BB|_{2^{n+1}-1}. \quad (9)$$

$$x_4 = |X|_{2^{2n}-3}$$
$$= ||B_1|_{2^{2n}-3} + |2^n B_2|_{2^{2n}-3} + |2^{2n} B_3|_{2^{2n}-3}|_{2^{2n}-3}$$
$$= |B_1 + 2^n B_2 + 3B_3|_{2^{2n}-3}$$
$$= |B + 3B_3|_{2^{2n}-3}. \quad (10)$$

$$x_5 = |X|_{2^{2n}+1}$$
$$= ||B_1|_{2^{2n}+1} + |2^n B_2|_{2^{2n}+1} + |2^{2n} B_3|_{2^{2n}+1}|_{2^{2n}+1}$$
$$= |B_1 + 2^n B_2 + (-B_3)|_{2^{2n}+1}$$
$$= |B + (-B_3)|_{2^{2n}+1}. \quad (11)$$

Where;

$B = B_1 + 2^n B_2 = B_1 \diamond B_2$, where $\diamond$ is defined as B_1 concatenation B_2

$$= \underbrace{\underbrace{B_{2,n-1} \dots B_{2,0}}_{n} \underbrace{B_{1,n-1} \dots B_{1,0}}_{n}}_{2n}. \quad (12)$$

$$BB = \underbrace{B_{3,1} B_{3,0} B_{3,n} \dots B_{3,2}}_{n+1}. \quad (13)$$

From Fig. 2, a given number X is partitioned into three-bit blocks (as in Equation 4) using a bits partitioning unit. Residues x_1 is computed using an $n\ bit$ wide Carry Save Adder to generate a partial sum s_1 and a carry c_1 which are then added using another $n\ bit$ wide Carry Propagate Adder. The residue x_2 from Equation (8) represents the n least significant bits of the binary form of X, which is equivalent to B_1. Therefore, computing requires no hardware utilization. Residues x_3 is computed by adding the results of concatenation and shift operations (both of which requires no additional hardware use) using a $n + 1\ bit$ Carry Propagate Adder. The redundant residues x_4 and x_5 are computed by a concatenation operation and a modulo addition operation (requiring $2n\ bit$ and $2n + 1\ bit$ wide CPAs). From the preceding, the Area of the proposed forward converter is estimated at $(7n + 2)\Delta_{FA}$. Whiles the Delay is estimated at $(4n + 2)t_{FA}$ since the implementation is done in parallel.

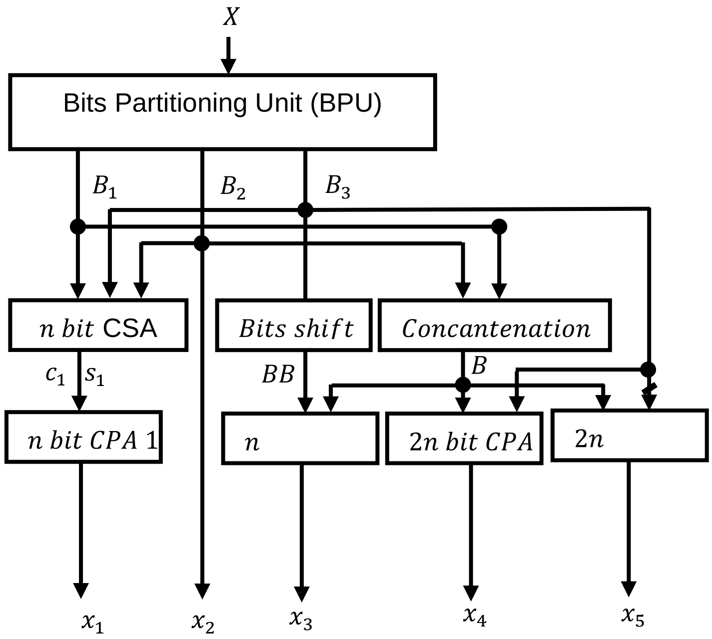

Fig. 2. Forward conversion architecture for the proposed scheme

3.1.1 Numerical example

Given the information moduli set, $\{2^n - 1,\ 2^n, 2^{n+1} - 1\}$ and a sense attribute value (given in integer) $X = 23$, the forward conversion process is as follows;

Let $n = 2$. Consequently, the moduli set is reduced to $\{3,\ 4,\ 7\}$.

Next 23 is converted into binary representation which gives 0010111_2. Since $n = 2$ and X is $3n + 1$ bits wide number, we partition X into two 2-bits blocks and one 3-bit block. Thus, $B_1 = 11, B_2 = 01\ and\ B_3 = 001$.

From Equation (7),

$$x_1 = |B_1 + B_2 + B_3|_{2^n-1}$$

$$= |(11)_2 + (01)_2 + (001)_2|_3$$

$$= |3 + 1 + 1|_3$$

$$= 2$$

From Equation (8),

$$x_2 = |B_1|_{2^n}$$

$$= |(11)_2|_4$$

$$= |3|_4$$
$$= 3$$

From Equation (9),

$$x_3 = |B + BB|_{2^{n+1}-1}$$

Where, from equation (12) and (13),

$$B = \underbrace{\underbrace{B_{2,n-1} \dots B_{2,0}}_{n} \underbrace{B_{1,n-1} \dots B_{1,0}}_{n}}_{2n}$$

$$= 0111$$

$$BB = \underbrace{B_{3,1}B_{3,0}B_{3,n} \dots B_{3,2}}_{n+1}$$

$$= 010$$

Therefore,

$$x_3 = |(0111)_2 + (010)_2|_7$$

$$= |7 + 2|_7$$

$$= 2$$

3.2 Reverse conversion

The front end of any RNS implementation involves forward conversion as elucidated in the immediate section above. Similarly, there is the need to perform reverse conversion of an encoded RNS data at the back end in order to make meaning out of what was transmitted. The reverse conversion process enables the legitimacy of a transmitted data to be tested, allowing afterwards error detection and correction if the data is in error. The reverse conversion process for the moduli set $\{2^n - 1,\ 2^n,\ 2^{n+1} - 1,\ 2^{2n} - 3,\ 2^{2n} + 1\}$, is performed using the traditional CRT as in Equation (2).

3.3 Proposed error detection and correction algorithm

The proposed scheme in this work uses the traditional CRT and the concept of Manhattan distance heuristics. The reasoning for using the CRT over MRC is that MRC uses mod-m_i for computation in a sequential fashion, which limits speed of operations [36]. Taking into consideration the limited battery reserve of sensor nodes, a worth choice is the CRT even though admittedly computation using CRT for reverse conversion may involve mod-M (which may be large). Importantly, to minimize such impact, the deployment of the reverse conversion process will be done at the Base station which is resource unlimited.

For any given moduli set $\{m_i\}$ for $i = 1, \dots, k,$ the equivalent decimal number X can be calculated from the residues $(x_1,\ x_2, \dots, x_k)$ using the CRT as in Equation (2).

Additionally, we employ the concept of heuristics in deciphering error bits and recovering an intended transmitted data. Specifically, the Manhattan distance heuristic will be used. A heuristic function $h(x)$ defined in equation (14) sums the distance that the vector x is far apart from the received vector τ. In other words, it measures the goodness of the vector x. The distance is measured by the sum of the differences in the x-positions and the y-positions of both vector x and vector τ (the received data). The y-positions are taken to be 0.

$$h(x) = D^M(x, \tau) \tag{14}$$

where;

$$D^M(x,\tau) = \sum_{i=1}^{n} abs(x_i - \tau_i) \quad (15)$$

Theorem 1: A code Ω based on a redundant residue number system can correct up to t errors; $t \leq \lfloor r/2 \rfloor$ where $\lfloor * \rfloor$ is the largest integer less than or equal to $*$ [16].

Theorem 2: Given the code $RRNS\ (n,k)$, such that no more than $2t$ errors have appeared in the received vector τ, the original integer X can be restored by iteratively performing Equation (16) to find the one value of X that is legitimate and with the minimum Manhattan distance.

$$X = R_\tau mod\ M_\tau \quad (16)$$

where, R_τ is the magnitude of the reduced residue after $2t$ possible error residues have been deleted from τ. $M_\tau = \prod_{\tau_j=1}^{n-2t} m_{\tau_j}$, is the magnitude of the corresponding moduli m_τ of residue positions without error.

Proof:

Consider the $RRNS\ (n,k)$ code such that up to t errors can be corrected. Let an integer X in the range $[0, M_K)$ have the residue vector $= \{x_i, \ldots, x_n\}$. If $2t$ errors are propagated into the vector x during transmission then a possible erroneous vector τ received at the destination which differs from x at $2t$ residues positions is given as;

$$\{\tau_i\}_{i=1,n} = \{x_i\}_{i=1,n-2t} + \{e_{u_i}\}_{u_i=1,2t} \quad (17)$$

In decimal form we can rewrite Equation (17) as;

$$Ɽ = (X + E) modM \quad (18a)$$

Which is equivalent to;

$$X = (Ɽ - E) mod \quad (18b)$$

Where X, Ɽ, and E are calculated using Equation (2).

It is clear from Equation $(18b)$ that if an $RRNS\ (n,k)$ code is able to detect $2t$ residues errors, then, the remaining residues and their corresponding moduli should be enough to recover the original integer X as long as it is not less than $k = n - r$. Which proofs that X can be generated from Equation (16).

It is worth noting also that, given that vector $\{\tau_i\}_{i=1,n}$ is the super set of $\{e_{u_i}\}_{u_i=1,2t}$, the proposed algorithm generates n different combinations of $r_\tau s$ from the complement of e.

Therefore,

$$\{r_{\tau_i}\}_{i=1,n-2t} = \{e_{u_i}\}'_{u_i=1,2t} \quad (19)$$

Each r_τ together with their respective $m_\tau's$ are used to generate n different R_τ using Equation (2).

Later, it is demonstrated numerically, that there is one and only one legitimate R_τ with the smallest Manhattan distance and represents the recovered integer.

The error detection and correction process involves solving Equation (16) iteratively to find the combination of R_{τ_j} and M_{τ_j} that yields a legitimate X and returns the smallest D^M; j is one of the combinations of n possible combinations. The steps involved in the multiple error detection and correction are presented in the algorithm below.

Algorithm: *errorHandlingInWSN*

1. Decode $\boldsymbol{Ɽ}$ from its encoded vector $\boldsymbol{\tau}$ using Equation (2)
2. IF $\boldsymbol{Ɽ}$ is in the legitimate range, output $\boldsymbol{Ɽ}$ as $\boldsymbol{X}$ and $GOTO\ END$
   // $Ɽ$ is the original message without error.
3. ELSE
4. SET $\boldsymbol{R_\tau} = \{\}; \boldsymbol{j} = 1$
5. WHILE $\boldsymbol{j} \leq \boldsymbol{n}$ DO
   COMPUTE $\boldsymbol{R_{\tau_j}}$ *from Equation (16)*
   IF $\boldsymbol{R_{\tau_j}}$ *is in the legitimate range*
   SET $\boldsymbol{R_\tau} \leftarrow ADD(\boldsymbol{R_{\tau_j}})$
   $j++$*; GOTO 5*
6. END WHILE
7. IF $\boldsymbol{R_\tau}.length() = 1$
8. RETURN $\boldsymbol{R_{\tau_1}}$ as $\boldsymbol{X}$ and $GOTO\ END$
9. *ELSE*
10. Calculate the residue vectors r_{τ_j} for each
    $\boldsymbol{R_{\tau_j}} \in \boldsymbol{R_\tau}$ and their $\boldsymbol{D^M}(r_{\tau_i}, \ \tau)$
11. RETURN $\boldsymbol{R_{\tau_j}}$ with the minimum $\boldsymbol{D^M}(r_{\tau_i}, \ \tau)$
    $min = h(r_{\tau_1})$
    $WHILE\ j \leq \boldsymbol{R_\tau}.Length()$
    $if\ h\left(r_{\tau_j}, \ \tau\right) \leq min$
    $min = h(r_{\tau_j})$
    END
    $RETURN\ \boldsymbol{R_{\tau_j}}\ as\ \boldsymbol{X}$
12. RETURN $2t$ error (s) found and t error (s) corrected.
13. END

3.4 Numerical Examples

Numerical examples are present for both single error and multiple error detection and correction based on the proposed error correction approach in here.

3.4.1 Single error detection and correction

The proposed moduli set $\{2^n - 1, \ 2^n, \ 2^{n+1} - 1, \ 2^{2n} - 3, 2^{2n} + 1\}$ will be considered. This generates a $RNS\ (5, 3)$ code which from Theorem 1, can detect 2 errors and correct 1 error.

If we consider $n = 2$, then we have the resulting moduli set $\{3, 4, 7, 13, 17\}$. The legitimate range is $[0, 84)$; while the illegitimate range is [84, 18564).

Let $X = 23$ and the equivalent residue vector $x = \{2, 3, 2, 10, 6\}$. Assume that a single error ($t = 1$) have been propagated into X during transmission at the 2nd residue position from 3 to 1. Then the erroneous received vector $\tau = \{2, 1, 2, 10, 6\}$.

Given $\tau = \{2, 1, 2, 10, 6\}$, and the moduli set $\{3, 4, 7, 13, 17\}$, the corresponding received decimal value Ɽ is computed using Equation (2).

The corresponding values of M_i and the multiplicative inverse M_i^{-1} are computed below;

M_i= {6188, 4641, 2652, 1428, 1092}; $i = 1, \dots, n$.

M_i^{-1} = {2, 1, 6, 6, 13}; $i = 1, \dots, n$.

Thus;

$$\text{Ʀ} = (2(6188)(2) + 1(4641)(1) + 2(2652)(6) + 10(1428)(6) + 6(1092)(13))\ mod\ 18564$$
$$= (24752 + 4641 + 31824 + 85680 + 85176) mod\ 18564$$
$$= (232073)\ mod\ 18564 = 9305$$

Since the calculated value of Ʀ $= 9305$ is an illegitimate number in the given RRNS code, it can be concluded that there is 1 error (in this illustration). In this case, Equation (16) is performed iteratively in $j \leq 5$ recovery trials. All combinations of $n - k$ (= 2) error residues are generated and the corresponding decimal values (R_{τ_j}) are calculated. Table 1 shows the results.

Table 1. Possible Error positions and corresponding weighted values

Iterations ($\boldsymbol{j}$)	Possible error positions		Reduced Vectors				$\boldsymbol{R_\tau}$
	$\boldsymbol{u_1}$	$\boldsymbol{u_2}$	$\hat{r}_{\tau_j}$	$\boldsymbol{m_{\tau_j}}$	$\boldsymbol{M_{\tau_j}}$	$\boldsymbol{R_{\tau_j}}$	$\boldsymbol{R_{\tau_j} mod\ M_{\tau_j}}$
1	1	2	$\{2, 10, 6\}$	$\{7, 13, 17\}$	1547	10852	23
2	2	3	$\{2, 10, 6\}$	$\{3, 13, 17\}$	663	8642	23
3	3	4	$\{2, 1, 6\}$	$\{3, 4, 17\}$	204	1145	125
4	4	5	$\{2, 1, 2\}$	$\{3, 4, 7\}$	84	149	65
5	5	1	$\{1, 2, 10\}$	$\{4, 7, 13\}$	364	2753	205

There are two possible results from Table 1 that are within the legitimate range of [0, 84). These are added to the set $\boldsymbol{R_\tau}$ forming $\boldsymbol{R_\tau} = \{23, 65\}$. The proposed algorithm breaks the tie by using the concept of Manhattan distance heuristic as in Equation (15). Table 2 shows the residues of $\boldsymbol{R_{\tau_1}}$ and $\boldsymbol{R_{\tau_2}}$ w.r.t the moduli set $\{3, 4, 7, 13, 17\}$, and the Manhattan distances from the received vector $\tau = \{2, 1, 2, 10, 6\}$.

The algorithm returns the integer value of $\boldsymbol{R_{\tau_i}}$(or r_{τ_i}) that has the smallest Manhattan distance from the received vector τ. Therefore, the recovered integer is 23 since it has the smallest Manhattan distance value of 2. The vector positions of 23 that have non-zero Manhattan distances from those of τ are in error. Hence, it is clear the error residue is at position 2; and the intended transmitted residue at the receiving sink node is; $|23|_4 = 3$.

Table 2. Residue vectors and Manhattan Distances for tie breaking

$\boldsymbol{i}$	$\boldsymbol{R_{\tau_i}}$	$\boldsymbol{r_{\tau_i}}$	$\boldsymbol{\tau}$	$\boldsymbol{D^M(r_{\tau_i}, \tau)}$
1	23	$\{2, 3,2,10,6\}$	$\{2,1, 2,10,6, \}$	2
2	65	$\{2,1, 2, 0,14\}$	$\{2,1, 2,10,6, \}$	18

3.4.2 Multiple error detection and correction

In order to test the error handling algorithm for multiple error detection and correction, the proposed moduli set $\{2^n - 1,\ 2^n,\ 2^{n+1} - 1,\ 2^{2n} - 3, 2^{2n} + 1\}$ together with additional moduli $\{2^{2n} + 3,\ 2^{2n} + 7\}$ to increase the redundant moduli will be considered. This generates a $RNS\ (7, 3)$ code which from Theorem 1, can detect and correct up to 4 and 2 errors respectively.

If we consider $n = 2$, then we have the set $\{3, 4, 7, 13, 17, 19, 23\}$. The legitimate range is $[0, 84)$; while the illegitimate range is [84, 8112468).

Let $X = 55$ and the equivalent residue vector is $x = \{1,\ 3,\ 6,\ 3,\ 4,\ 17,\ 9\}$. Assume that two errors ($t = 2$) have propagated into X during transmission at the 3rd and 6th positions respectively. Let the received vector $\tau = \{1,\ 3,\ 11,\ 3,\ 4,\ 2,\ 9\}$.

Given $\boldsymbol{\tau} = \{1, 3, \mathbf{11}, 3, 4, \mathbf{2}, 9\}$, and the moduli set $\{3, 4, 7, 13, 17, 19, 23\}$, the corresponding received decimal value Ʀ is computed using Equation (2) as follows.

The corresponding values of M_i and the multiplicative inverse M_i^{-1} are computed below;

M_i= {2704156, 2028117, 1158924, 624036, 477204, 426972, 352716}; $i = 1, n.$
M_i^{-1} = {1, 1, 2, 4, 11, 5, 21}; $i = 1, n.$

Thus;

$$Ʀ = (1(2704156)(1) + 3(2028117)(1) + 11(1158924)(2) + 3(624036)(4) + 4(477204)(11) + 2(426972)(5) + 9(352716)(21)) \ mod\ 8112468$$
$$= (2704156 + 6084351 + 25496328 + 7488432 + 20996976 + 4269720 + 66663324) mod\ 8112468$$
$$= (133703287)\ mod\ 8112468 = 3903799$$

Since the calculated value of Ʀ $= 3903799$ is an illegitimate number in the given RRNS code, it can be concluded that there are errors (at least one). In this case, Equation (16) is performed iteratively in $j \leq 7$ trials. All combinations of $n - k$ $(= 4)$ error residues are generated and the corresponding decimal values (R_{τ_j}) are calculated. Table 3 shows the results.

There are two possible results from Table 3 that are within the legitimate range of [0, 84). These are added to the set $\boldsymbol{R_\tau}$ resulting in $\boldsymbol{R_\tau} = \{55, 67\}$. The proposed algorithm breaks the tie by using the concept of Manhattan distance heuristic as in equation (15). Table 4 has the residues of $\boldsymbol{R_{\tau_2}}$ and $\boldsymbol{R_{\tau_6}}$ and Manhattan distances from the received vector $\tau = \{1,3,4,3,4,2,9\}$.

Table 3. Possible Error positions and corresponding weighted values

No of iterations	Possible Error Positions				Reduced Vectors				$\boldsymbol{R_\tau}$
$\boldsymbol{J}$	$\boldsymbol{u_1}$	$\boldsymbol{u_2}$	$\boldsymbol{u_3}$	$\boldsymbol{u_4}$	$\hat{r}_{\tau_j}$	$\boldsymbol{m_{\tau_j}}$	$\boldsymbol{M_{\tau_j}}$	$\boldsymbol{R_{\tau_j}}$	$\boldsymbol{R_{\tau_j} mod\ M_{\tau_j}}$
1	1	2	3	4	{4, 2, 9}	{17,19,23}	7429	2692872	3574
2	3	4	5	6	{1, 3, 9}	{3,4,23}	276	2439619	55
3	5	6	7	1	{3,6,11}	{4,7,13}	364	6619239	263
4	7	1	2	3	{11, 4,2}	{13,17,19}	4199	305256	2928
5	2	3	4	5	{2,9,1}	{19,23,3}	1311	624988	952
6	4	5	6	7	{1,3,6}	{3,4,7}	84	1834963	67
7	6	7	1	2	{6, 11,4}	{7,13,17}	1547	5306928	718

Table 4. Residue vectors and Manhattan Distances for tie breaking

$\boldsymbol{i}$	$\boldsymbol{R_{\tau_i}}$	$\boldsymbol{r_{\tau_i}}$	$\boldsymbol{\tau}$	$\boldsymbol{D^M(r_{\tau_i}, \tau)}$
1	55	{1, 3,6,3,4,17,9}	{1, 3,4,3,4,2,9}	17
2	67	{1, 3, 4, 2, 16, 10,21}	{1, 3,4,3,4,2,9}	33

The algorithm returns the value of $\boldsymbol{R_{\tau_i}}$ that has the smallest Manhattan distance from received vector τ. Therefore, the recovered integer is 55 since it has the smallest Manhattan distance value of 17. The vector positions of 55 that have non-zero Manhattan distances from those of τ are in error. Hence, it is clear the error residues are at position 3 and 6; and the intended transmitted residues are derived as; $|55|_7 = 6$ and $|55|_{19} = 17$.

Notice from above that the error correction and detection process was achieved in 7 trials (equivalent to the code size). But, in the best-case scenario the proposed algorithm could achieve the error correction in lesser than n trials.

4 Results and Performance Comparison

The performance of the proposed error detection and correction algorithm is evaluated primarily in terms of computational time - number of iterations needed to recovers a message. Compared to exiting schemes [21,27,37] and [29], the scheme inhere offers fewest iterations to recover an original transmitted. In the worst-case scenario, the algorithm in this work requires n (the codes length) trials to decode an integer value equivalent of a signal that has been transmitted over a sensing network. Table 5, gives a comparative analysis of performance of the proposed scheme and recent schemes.

A careful look at Table 5, show the most recent error correction method presented by [21] requires C_t^n trials to detect and correct both single and multiple errors that may be contained in a transmitted data. This is same as the works of [16] for single error correction; but better than the schemes presented in [37,29] and [27]. The proposed scheme performs better than these schemes with only n trials required in the worst-case scenario to achieve the function of detecting and correcting multiple errors. Fig. 3 below compares the error handling scheme in [21] and the proposed scheme for $r = 4$ and varying k. The scheme in [21] presents a single hardware implementation for both forward and reverse conversion; while the proposed scheme here provides only an architecture for forward conversion. It will be worthwhile to build a combined reverse conversion architecture for the moduli set here in order to measure and compare the efficiency of the hardware implementations of these divergent approaches to detecting and correcting errors.

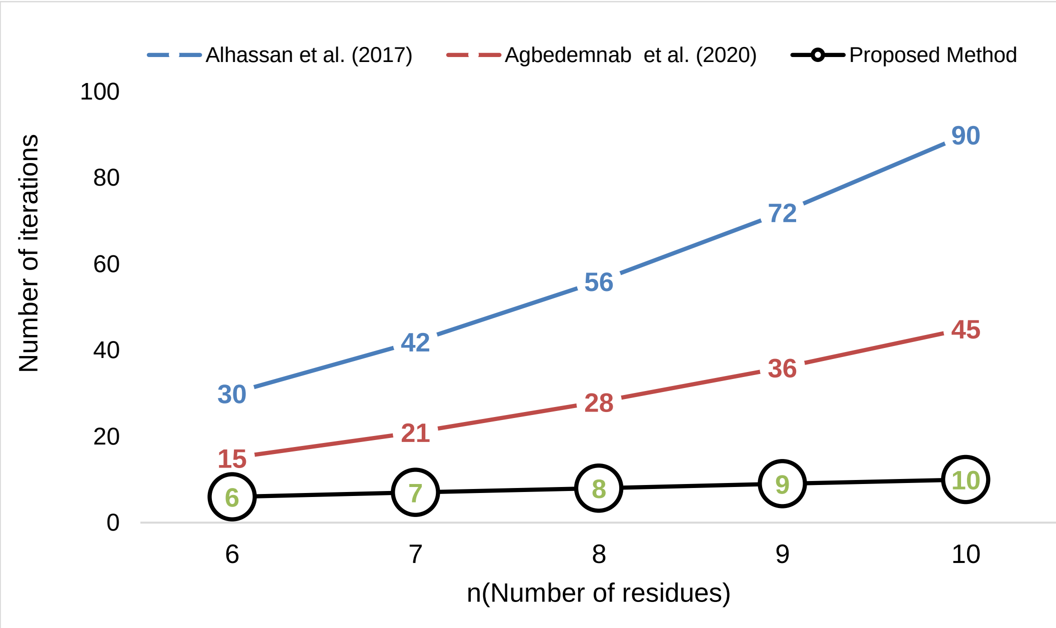

Fig. 3. Comparison of error recovery times for various schemes

Table 5. Performance comparison of error detection and correction schemes

Scheme	Integer Recovery Method	Iterations	Error Type		Converter Availability	
		Worst Case	Single	Multiple	Forward	Reverse
Salifu and Gbolagade [29]	MRC	$2 \times C_t^n$	Yes	No	Yes	No
Karthik, et al. [37]	CRT	C_k^n	No	Yes	No	No
Alhassan, et al. [30]	CRT	$2 \times C_t^n$	Yes	No	Yes	No
Agbedemnab, et al. [21]	CRT	C_t^n	Yes	Yes	Yes	Yes
Proposed Scheme	CRT	n	Yes	Yes	Yes	No

5 Conclusion

Reliable data transmission through communication systems such as wireless sensor networks are vital for important application-specific decision making regarding human life and properties. In this work a new algorithm for single and multiple residue error detection and correction has been proposed. The proposed algorithm which uses redundant residue number coding and the Chinese Remainder Theorem stands out amongst recent algorithms in terms of the number of trials required to correct errors. A new concept of Manhattan distance heuristics is used to recover erroneous digits where there is tie of legitimate received values. This heuristic not only makes it easier to recover the erroneous integer but also easier to decode the erroneous digit positions much faster. In future, the algorithm will be tested on a routing protocol for WSN to simulate the benefits in terms of increased throughput and reduction in energy consumption.

Acknowledgement

The authors acknowledge the valuable contributions and technical assistance of reviewers.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Pottie G, Kaiser W. Wireless integrated network sensors. Communication of ACM. 2000;43(5):51–58.

[2] Tay T, Chang CH, Low J. Efficient VLSI implementation of 2n scaling of signed integer in RNS (2n−1, 2n, 2n+1). IEEE Trans. Very Large Scale Integr. Syst. 2013;21:1936–1940.

[3] Jenkins WK, Altman EJ. Self-checking properties of residue number error checkers based on mixed radix conversion. IEEE Transactions on Circuits and Systems. 1988;35(2):159–167.
Available:https://doi.org/10.1109/31.1717

[4] Xiao H, Garg HK, Hu J, Xiao G. New error control algorithms for residue number system codes. ETRI Journal. 2016;38(2):326–336.
Available:https://doi.org/10.4218/etrij.16.0115.0575

[5] Szabo and Tanaka. Residue arithmetic and its application to computer technology. McGraw-Hill Book Co.; 1967.

[6] Claudio EDD, Orlandi G, Piazza F. A systolic redundant residue arithmetic error correction circuit. IEEE Trans. Comput. 1993;42(4):427–432.

[7] Cheney PW. An investigation of residue number theory for digital systems, Ph. D. dissertation. Stanford Univ., Calif; 1962.

[8] Watson RW, Hastings CW. Self-checked Computations using Residue Arithmetic: Proc. of IEEE. 1966;54(12):1920-1931.

[9] Yau SS, Liu YC. Error correction in RRNS: IEEE trans. Computer. 1973; c-22:5-11.

[10] Mandelbaum DM. Error–correction in Residue Arithmetic, IEEE Trans. Computer. 1972;21(6):538-545.

[11] Barsi F, Maestrini P. Error correcting properties of redundant residue number systems. IEEE Transaction on Computers; 1973.

[12] Tang Y, Boutillon E, Jégo C, Jézéquel M. A new single-error correction scheme based on self-diagnosis residue number arithmetic. Design and Architecture for Signal and Image Processing Conf., Edinburgh, Scotland. 2010;27(33).

[13] Ramachandran V. Single residue error correction in residue number systems, IEEE Trans. Computers. 1983;c-32(5):504-507.

[14] Su CC, Lo HY. An algorithm for scaling and single residue error correction in residue number systems. IEEE Trans. Comput. 1990;39(8):1053-1064.

[15] Katti RS. A new residue arithmetic error correction scheme. IEEE Trans. Comput. 1966;45(1):13-19.

[16] Goh VT, Siddiqi MU. Multiple error detection and correction based on redundant residue number systems. IEEE Trans. Commun. 2000;56(3):325–330.

[17] Krishna H, Sun JD. A coding theory approach to error control in RRNS-part 1: - Theory and single error-correction. IEEE Trans. Circuits and Syst. II Analog & DSP. 1992;39:8-17.

[18] Amusa KA, Nwoye EO. Novel Algorithm for Decoding Redundant Residue Number Systems (RRNS) Codes; 2012.

[19] Gbolagade KA, Cotofana SD. Generalized matrix method for efficient residue to decimal conversion. IEEE Trans. Computer. 2008;1414 –1417.

[20] Yang LL, Hanzo L. Redundant residue number system based error correction codes. Vehicular Technology Conference, VTC Fall. 2001;3:1472 -1476.

[21] Agbedemnab PA, Baagyere EY, Daabo MI. Single and multiple error detection and correction using redundant residue number system for cryptographic and stenographic schemes. Asian Journal of Research in Computer Science. 2019;4(4):1-14.

[22] Roshanzadeh M, Saqaeeyan S. Error detection & correction in wireless sensor networks by using residue number systems. International Journal of Computer Network and Information Security. 2012;4(2):2935.
ISSN: 20749090.
DOI: 10.5815/ijcnis.

[23] James J, Ameenudeen P. A novel method for error correction using redundant residue number system in digital communication systems. International Conference on Advances in Computing, Communications and Informatics (ICACCI); 2015.

[24] Zarei B, Muthukkumarasamy V, Wu XW. Energy efficient channel coding of physiological signals in wireless body area networks. International Journal of Computer Applications. 2015;0975 –8887:118(23)

[25] Alhassan AB, Saeed I, Agbedemnab PA. The Huffman's method of secured data encoding and error correction using residue number system (RNS). Communication on Applied Electronics (CAE) Journal, Foundation of Computer Science (FCS). 2015;2(9).
ISSN: 2394-4714.

[26] Alhassan AB, Gbolagade KA, Bankas EK. New Lempel-Ziv-Welch fault tolerant data compression and encryption scheme. International Journal of Advanced Studies in Engineering and Scientific Intervention. 2017a;4(1).

[27] Alhassan AB, Gbolagade KA, Bankas EK. A Novel and efficient LZW-RNS scheme for enhanced information compression and security. International Journal of Advanced Research in Computer Engineering and Technology. 2017b;4 (11):4015-4019.

[28] Olatunde OT, Gbolagade KA, Abolaji Y. Redundant Residue Number System Based Fault Tolerant Architecture over Wireless Network; 2013.

[29] Salifu AM, Gbolagade KA. An improved redundant residue number system-based error detection and correction scheme for the moduli set. 2016;2(1):1114.

[30] Alhassan AB, Daabo MI, Akobre S. Improved Lempel-Ziv-Welch's Error Detection and Correction Scheme using Redundant Residue Number System (RRNS). Circulation in Computer Science. 2017;2(6): 25-30.

[31] Bankas E, Gbolagade K. New Efficient FPGA Design of Residue to Binary Converter. International Journal of VLSI design & Communication Systems (VLSICS). 2013;4:6.

[32] Gbolagade KA. An Efficient MRC based RNS-to-Binary Converter for the {22n−1,2n,22n+1−1} Moduli Set. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). 2013;2(4).

[33] Pontarelli S, Cardarilli GC, Re M, Salsano A. A novel error detection and correction technique for RNS based FIR filters,'' IEEE Int. Symp. Defect and Fault Tolerance of VLSI Systems, Boston, MA. 2008; 436-444.

[34] Omondi A, Premkumar B. Residue number systems: Theory and implementation. Imperial College Press, UK; 2007.

[35] Taylor J. Residue arithmetic: A tutorial with examples. Computer. 1984;17(5).

[36] Gbolagade K, Chaves R, Sousa L, Cotofana S. An improved RNS reverse converter for the {22n+1 - 1, 2n, 2n - I} moduli set. IEEE International Symposium on Circuits and Systems, Paris, France, Pages. 2010;2103-2106.

[37] Karthik G, Mohan Raj R, Karthik B. RRNS based error detection and correction in CDMA using Chinese Remainder Theorem. 2016;2(2):338341.

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)
https://www.sdiarticle4.com/review-history/72725