RESIDUE NUMBER SYSTEM BASED APPLICATIONS: A LITERATURE REVIEW

Afeez Adeshina Oke[1], Babatunde Akinbowale Nathaniel[1],
Balogun Fatimah Bukola[1], Oloyede Abdulkarim Ayopo[2]

[1] **Department of Computer Science, Faculty of Communication and Information Technology, Kwara State University, Malete, Kwara State, Nigeria**
[2] **Department of Telecommunication, Faculty of Communication and Information Sciences, University of Ilorin, Kwara State, Nigeria**

Corresponding author: Babatunde Akinbowale, akinbowale.babatunde@kwasu.ed.ng

ABSTRACT: Residue Number System (RNS) has become one of the most preferred solutions for the implementation of distributed and ubiquitous computing platforms such as cloud, wireless adhoc networks, applications which require tolerance against errors and scalable solutions in mission critical and next generation's applications. The implementation of RNS for different applications require different approaches depending on the areas of applications. Some applications of Residue Number System described in literature are reviewed so as to illustrate the various· possibilities. Distinct features such as applications, architectures and implementations were considered. In this paper, we presented a comprehensive survey on the different areas of applications, architectures and implementations of RNS to various fields of Information Technology while also including new areas of applications hitherto not covered in previous surveys. The implementations issues with different types of applications such as moduli set, forward conversion, residue arithmetic units, reverse conversion and hardware design, were discussed. Lastly, we focus on the various challenges with the use of RNS and the different solutions that exist and also discuss the future trends in RNS.

KEYWORDS: Moduli set, forward conversion, reverse conversion, Chinese Remainder Theorem, Mix Radix Conversion

1. INTRODUCTION

Number representation in a digital system affects and impacts all levels of design abstraction from hardware architecture to algorithms (Chang, 2015). The type of number system for the hardware implementation of an application impacts its workload by imposing the number and complexity of operations required to accomplish different specific tasks. Additionally, parallelization of algorithms is hugely dependent on the type of number representation. The introduction of Residue Number Systems (RNS) commences a paradigm shift in number representation. This shift divides a weighted number system by a set of moduli in order to achieve residues(Aremu & Gbolagade, 2017; N. Singh, 2016a). This is done by forward converter, then, arithmetic operations are done on residues instead of initial weighted number by independent modulo arithmetic units. This results in performing operations in parallel by omitting carry propagation between them, and therefore having high-speed addition, subtraction and multiplication. Finally, the reverse converter decodes the resulted residues to the corresponding weighted.

The properties of RNS has allowed researchers apply the number system for high speed arithmetic's, fault tolerant systems due to no error propagation and complex number arithmetic (Aremu & Gbolagade, 2017; N. Singh, 2016a). All these features increase the scientific tendency toward the RNS especially for digital signal applications and cryptography. Advantages of RNS will be further demonstrated from this perspective by new applications such as multicarrier CDMA for broadband mobile communication systems, reliability enhancement in wireless sensor networks, tableless routing in software defined networks, packet processing and routing for mobile ad hoc network, security in cloud computing and block chain technology, digital watermarking, minimizing hardware cost of convolutionary neural networks etc. However, the RNS is still not popular in implementations due to the complexities of some operations such as magnitude comparison and division operations (Omondi & Premkumar, 2007), overhead associated with conversions (Somayah, Mahmood, & Sorin, 2010) and the complex nature of the residue-to-binary converter architecture (Gbolagade, 2009).

This paper presents the existing RNS applications and solutions in implementing RNS based systems. The review also covered an in-depth description of RNS with examples noting the factors mitigating against the widespread use of the number system. Additionally, most surveys on RNS has majorly focused on the operations of RNS (Aremu & Gbolagade, 2017; Navi, Molahosseini, & Esmaeildoust, 2011; N. Singh, 2016a). Researchers have investigated the applications of RNS to different applications however, most of the reviews have been limited to related applications (Eseyin, 2019; Mohan, 2002; D. Schinianakis & Stouraitis, 2016; Shrimali & Sharma, 2018; Soderstrand, Jenkins, Jullien, & Taylor, 1986). This survey focuses on different areas of applications of RNS while also including new areas of applications hitherto not covered in previous surveys. With the new areas of applications comes new challenges with the use of RNS, we focus on this challenges and further discuss different solutions that exist and also discuss the future trends in RNS.

In the next section of the paper, the fundamental concepts of RNS, including the common notations, definitions and general architecture and merits and demerits, are introduced. The different RNS based applications are described in Section III. The aim is to present these applications in a way that will motivate the effective use of RNS to improve these applications and develop new RNS based applications for new domain specific computing. Section IV discusses the challenges and opportunities of implementing RNS-based computations. Finally, the paper is concluded in Section V with an anticipated future of RNS for new applications.

2. RNS BACKGROUND

All digital systems design depends so much on the number system (Abdul-Barik, 2016). A number system is a writing system for describing or expressing numbers. A number system consists of a correspondence between sequences of digits and numbers. In a fixed-point number system, each sequence corresponds to exactly one number (Soderstrand, Jenkins, Jullien, & Taylor, 1986). There are basically two types of number system, the conventional number system also called the weighted or positional number system (such as binary, decimal and octal number systems) and the unconventional number system also called the unweighted or non-positional number system (such as the gray code and residue number system). The latter is called a non-positional arithmetic because positional information is always lost after any arithmetic operation with the number systems. Digital systems are mostly built around the weighted number system (WNS) (Abdul-Barik, 2016) whereas the major challenge of WNS is with the carrying propagation chains in it's operations and to improve the performance of processors built around WNS in terms of speed and area cost there is need to eliminate the associated inherent carry propagation (Gbolagade 2010) .

Residue Numbers System (RNS) is an alternative in this regard as it splits a large number into smaller ones such that arithmetic operations are performed on smaller numbers rather than the original large number thereby removing totally carry propagation problems in addition, subtraction and multiplication. RNS is a non-positional number system which speeds up arithmetic operations by splitting numbers into smaller parts in such a way that each unit is independent of the other and arithmetic operations are carried out on these smaller parts at the same time rather than on the original number (Gbolagade, 2010; Gbolagade & Cotofana, 2009; Rooju, 2008; Alhassan, 2013; Chaifali, Partha & Amitabha, 2001). It is based on the congruence relation which explains that two integers a and b are said to be in congruent modulo m if m divides exactly the difference of a and b; mathematically represented as $a \equiv b(mod\ m)$. E.g $640 \equiv 331(mod\ 3)$ (Omondi & Premkumar, 2007). If q and r are the quotient and remainder respectively of integer division of a by m that is a=q.m+ r then by definition $a \equiv r(mod\ m)$ where $r = |a|_m$, that is, r is the residue of a with respect to m. The number m is a modulus or base and we shall assume that its value excludes unity which produces only trivial congruencies (Alhassan, 2013).

The history of Residue Number System (RNS) can be linked to a verse from a third century book written by a Chinese scholar Sun Tzu who posed a mathematical riddle with the following statements (Alhassan, 2013; Baagyere, 2011; Baagyere, Boateng & Gbolagade, 2011; Gbolagade, 2010; Gbolagade & Cotofana, 2009; Rooju G., 2008; Omondi & Premkumar, 2007; Chaifali, Partha & Amitabha, 2001; Yassine & Moore, 1991; Szabo, et. al, 1967):

> "We have things we do not know the number;
> If we count them by 3's, we have two left over
> If we count then by 5's, we have three left over
> If we count them by 7's, we have two left over
> How many things are there? "

This riddle means which number yields remainder 2, 3, 2 when divided by 3, 5, and 7 respectively.

Sun Tzu (1247) gave the solution to this puzzle called the Taiyen (Great Generalization) which gave the answer as 23. The Taiyen was later in 1247 generalized to what is now called the Chinese Remainder Theorem (CRT) by a Chinese mathematician (Qin Jiushao) (Alhassan, 2013; Baagyere, 2011; Baagyere, et. al, 2011; Gbolagade, 2010; Omondi & Premkumar, 2007).

RNS is defined in terms of a relatively prime moduli set $\{m_1, m_2, m_3, \ldots, m_n\}$ such that $\gcd(m_i, m_j) = 1$ for $i \neq j$, where gcd means the greatest common divisor m_i and m_j while $M = \prod_{i=1}^{n} m_i$, is the dynamic range. RNS is capable of uniquely representing all integer X that lie in its dynamic range, that is, $(0 \leq X < M)$ where the dynamic range (M) is determined by the multiplication of the moduli sets (Abdelfattah, 2011). If the result of a calculation exceeds M (Dynamic Range), then, an overflow has occurred.

The inherent and advantageous properties of this number system such as very fast arithmetic computations, parallel arithmetic operations, error detection and correction abilities etc has made scientists since in the 1950's put them to use in the implementation of fast arithmetic and fault tolerant computing (Alhassan, 2013; Baagyere, 2011; Gbolagade, 2010; Omondi & Premkumar, 2007).

RNS architectures are composed of three main parts; a binary –to – residue converter, residue arithmetic units and a residue – to – binary converter. This residue – to –binary converter is the most challenging part of any RNS architecture. However, for any successful application of RNS, data transformation from binary to residue and vice versa must be very fast so that conversion overhead does not nullify the advantage provided by the RNS (Gbolagade & Cotofana, 2009; Gbolagade, 2008; 2009). Data conversion and moduli selection are the two (2) most important issues for a successful RNS realization (Baagyere, et. al, 2011).

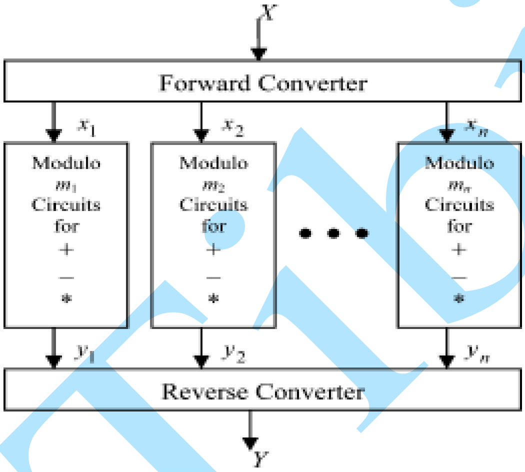

X
Forward Converter
x_1
x_2
x_n
Modulo m_1 Circuits for + – *
Modulo m_2 Circuits for + – *
Modulo m_n Circuits for + – *
y_1
y_2
y_n
Reverse Converter
Y

Figure 1.0 The General Structure of an RNS-based Processor

2.1 Data Conversion

In any RNS design, input operands are given in either binary or decimal format and must be converted to RNS notation before computation can be performed. The final results must be re-presented in the same way as the input operands. Hence, the need for a binary to residue, residue to binary converter for a successful RNS design is inevitable.

Two types of conversion are possible in a RNS processor; a forward conversion and a reverse conversion. The forward conversion mainly entails the transformation of a weighted integer number (decimal or binary number) into its equivalent RNS representation while the latter involves the translation of a residue represented number into its equivalent weighted number. The Forward conversion is a much simpler process and can be efficiently accomplished using multi-operand modular adders involves. The reverse conversion is one of the major concern in the full adoption of RNS due to its complex nature (Gbolagade, et. al, 2009; Aremu & Gbolagade, 2017; Deryabin, Chervyakov, & Tchernykh, 2018; Hiasat, 2019; Singh, 2016)). Major algorithms for performing cumbersome RNS operations, such as division and magnitude comparison, are based on the reverse conversion. Furthermore, it is a significant part of the RNS system because the conversion delay should not counteract the speed gain of the RNS arithmetic unit (Navi et al., 2011).

2.1.1 Forward Conversion

The forward conversion is performed by a forward converter which decomposes a weighted binary number into a residue represented number with regards to a moduli set, that is, it is the conversion from a conventional representation to a residue one by dividing the number X by each of the given moduli and then collect their remainder (Babatunde, 2019; Barik, 2016; Keivan, et. al, 2011; Omondi & Premkumar, 2007).

Thus, the residues of a conventional number X can be obtained as;

$x_i = |X|_{m_i}$……………………………..………. (2.0)

Example: Given the moduli set $\{2^n - 1, 2^n, 2^n + 1\}$ where $m_1 = 2^n - 1$, $m_2 = 2^n$ and $m_3 = 2^n + 1$ and n = 3 such that the moduli set is now $\{7, 8, 9\}$, the number 150 can be represented in RNS as;

$$x_1 = |X|_{m_1} = |150|_7 = 3;\ x_2 = |X|_{m_2} = |150|_8 = 6;\ x_3 = |X|_{m_3} = |150|_9 = 6$$

Thus, the RNS representation of 150 is thus $(3, 6, 6)_{RNS(7,8,9)}$.

The basic technique in converting binary to RNS was presented by (Babatunde, 2019; Barik, 2016; Baagyere, et. al, 2011; Omondi & Premkumar, 2007; Aremu & Gbolagade, 2017; Deryabin et al., 2018).

2.1.2 Reverse Conversion

The process of converting a residue arithmetic back to the conventional number (either binary or decimal) is referred to as reverse conversion. In the literature, the Chinese remainder theorem (CRT) and the Mixed Radix conversion (MRC) are the two (2) main methods for reversing residue arithmetic. All other developed or designed algorithms take their basis from these two (2) methods. These methods are either due to the type of moduli-set chosen or from working around some properties to suit the newly chosen approach (Omondi & Premkumar, 2007).

The Chinese remainder theorem is advantageous because its computations can be parallelized while that of the mixed radix conversion is sequential. However, the CRT is not being adopted by up to date reverse converters because of the following reasons:

i. The complex, large and slow modulo M operation(derived from the dynamic range) involved in its computation.
ii. The computation of the multiplicative inverse in its operations which has no general expression (formula) for its determination. A brute force search is the best method for its computation whereas not all residues have a multiplicative inverse. A residue with respect to a particular modulus has a multiplicative inverse iff the modulus is prime, that is, $|x^{-1}|_m$ exists only if x and m are relatively prime (Omondi & Premkumar, 2007).

However, before any method of the reverse converter can be used, the moduli set used must be coprime although there has been serious work on moduli sets with common factors (Gbolagade & Cotofana, 2009; 2008).

Mixed Radix Conversion (MRC) is an alternative to the CRT which does not involve the computation of large modulo-M in its operations. It involves a very large number of arithmetic computations which is due to the fact that the algorithm performs its operations sequentially (Gbolagade, 2010).

Schematically, we obtain:

2.1.2.1 Chinese Remainder Theorem

The Chinese Remainder theorem is defined as follows:

Given a pairwise relatively prime moduli set $\{m_1, m_2, \dots, m_k\}$ and a residue representation $(x_1, x_2, \dots, x_k)$, the magnitude of the residue can be obtained by using the equation.

$$X = \left|\sum_{i=1}^{n} xi.M_i\left|M_i^{-1}\right|_{mi}\right|_M \qquad (2.2)$$

Where $M_i = \frac{M}{m_i}$ and M_i^{-1} is the multiplicative inverse of M_i with respect to m_i such that $\left|M_i^{-1} * ?\right|_{mi} = 1$.

Such that

$$X = |m_1|M_1{}^{-1}|_{m_1}x_1 + m_2|M_2{}^{-1}|_{m_2}x_2 + m_3|M_3{}^{-1}|_{m_3}x_3 \dots \qquad (2.3)$$

To be able to use the CRT algorithm effectively, three (3) main steps are required:

i. The M_i's and their respective multiplicative inverse M_1^{-1}, M_2^{-1},……M_n^{-1} must be computed.
ii. Multiply and accumulate operations are followed.
iii. Modular reduction is performed.

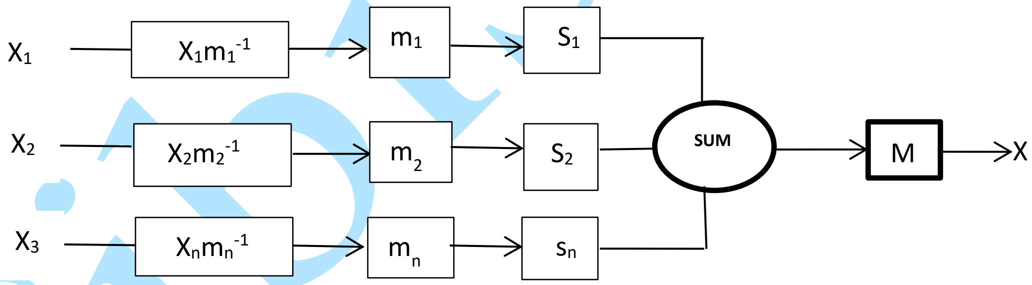


Figure 1.1 Schematic diagram for a Chinese Remainder Theorem

From the above schematic, it has been clearly shown that the CRT operations are parallelized (Baagyere, 2009).

Example: Consider the moduli set {3,4,5}. Compute the conventional number of the residue {2,3,1} with respect to the moduli-set using the traditional CRT.

Computing all M_i's : M_1= $M_1 = \frac{M}{m_1} = \frac{3*4*5}{3} = 20$

$$M_2 = \frac{M}{m_2} = \frac{3*4*5}{4} = 15$$

$$M_3 = \frac{M}{m_3} = \frac{3*4*5}{5} = 12$$

Computation of their respective multiplicative inverses: $\left|M_i^{-1} * ?\right|_{mi} = 1$

$$\left|M_1^{-1} * M_1\right|_{mi} = 1$$

$$\left|M_1^{-1} * 20\right|_3 = 1$$

$$|2 * 20|_3 = 1$$

$$M_1^{-1} = 2$$

Also,

$$\left|M_2^{-1} * M_2\right|_{mi} = 1$$

$$\left|M_1^{-1} * 15\right|_4 = 1$$

$$|3 * 15|_3 = 1$$

$$M_2^{-1} = 3$$

Also,

$$\left|M_1^{-1} * M_3\right|_{mi} = 1$$

$$\left|M_1^{-1} * 12\right|_5 = 1$$

$$|3 * 12|_5 = 1$$

$$M_3^{-1} = 3$$

Substituting the derived values and the moduli set into the traditional CRT $X = \left|\sum_{i=1}^{3} xi.M_i\left|M_i^{-1}\right|_{mi}\right|_{60}$ $= |(2 * 20 * 2) + (3 * 15 * 3) + (1 * 12 * 3)|_{60} = 11$.

Simplification of certain moduli sets are possible, the traditional Chinese remainder theorem can be simplified to derive several versions of the CRT. It is based on this advantage that several efficient reverse converters have been derived and reported in literature (Hiasat, 2017, 2019; Phalguna, Kamat, & Ananda Mohan, 2018; Phalguna, Kamat, & Mohan, 2019).

2.1.2.2 Mixed Radix Conversion

The mixed radix conversion which is an alternative to the Chinese Remainder theorem is defined as follows: Given a set of coprime moduli set {m_i, m_2, m_3,….., m_n} and a residue $(x_1, x_2, \dots, x_k)$, the decimal equivalent of the residue arithmetic can be computed as given in equation (1.3); (Nazarov & Chervyakov, 2018; Phalguna et al., 2018, 2019; Thabah, Sonowal, & Saha, 2018)).

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2 + \cdots + a_n m_1 m_2 m_3 \dots m_{k-1} \cdots$$

Where the Mixed Radix Digits (MRDs), a_i for i=1, 2,…..k can be calculated as:

$$a_1 = x_1$$

$$a_2 = \left| (x_2 - a_1) \left| m_1^{-1} \right|_{m_2} \right|_{m_2}$$

$$a_3 = \left| ((x_3 - a_1) \left| m_1^{-1} \right|_{m_3} - a_2) \left| m_2^{-1} \right|_{m_3} \right|_{m_3}$$

.

.

.

$$a_n = \left| (((x_k - a_1) \left| m_1^{-1} \right|_{m_k} - a_2) \left| m_2^{-1} \right|_{m_k} - \cdots - a_{k-1}) |m_{k-1}^{-1}|_{m_k} \right|_{m_k}$$

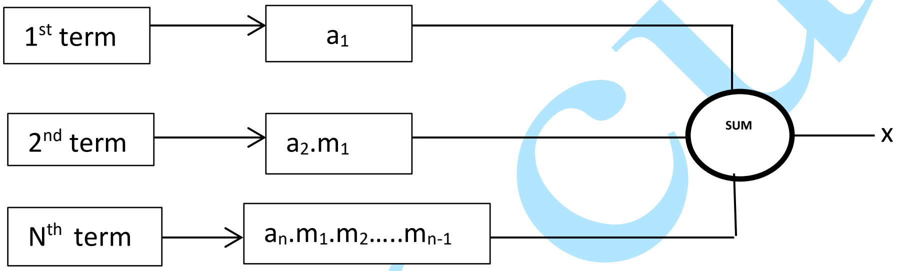


Figure 1.2: The schematic diagram of the MRC based on the above equation

For the MRDs a_i, $0 \leq a_1 < m_i$, any positive number in the interval [0, $\prod_{i=1}^{n} mi - 1$] can be uniquely represented. The major disadvantage of this method is that MRC is sequential in nature which makes CRT because of its parallel computation faster than the MRC.

Example:

Given the moduli set {3,4,5}. Compute the conventional number of the residue {2,3,1} with respect to the moduli-set using the Mixed Radix conversion.

First, we compute $| m_1^{-1} |_{m_2}$ as follows

$$|| m_1^{-1} |_{m_2} \; m_1 |_{m_2} = 1$$

$$|| m_1^{-1} |_{m_2} \; {*}3 |_4 = 1$$

$$| m_1^{-1} |_{m_2} = 3$$

Also,

$$| (m_2 m_1)^{-1} |_{m_3}$$

$$|| (m_2 m_1)^{-1} |_{m_3} \; {*}12 |_5 = 1$$

$$| (m_2 m_1)^{-1} |_{m_3} = 3$$

The values of a_1, a_2, and a_3 are obtained as follows:

$$a_1 = x_1 = 2$$

$$a_2 = || m_1^{-1} |_{m_2} \; (x_2 - a_1) |_{m_2}$$

$$= | 3 * (3 - 2) |_4$$

$$= 3$$

$$a_3 = || (m_2 m_1)^{-1} |_{m_3} \; (x_3 - (a_2 m_1 + a_1)) |_{m_3}$$

$$= | 3 * (1 - (3 * 3 + 2)) |_5$$

$$= 0$$

Therefore, the number X has the mixed-radix representation {2,3,0}

To obtain X in conventional form, we the radix digits to equation

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2$$
$$= 2 + (3 * 3) + (0 * 4 * 3) \qquad (2.4)$$
$$= 11$$

Thus, 11 is the value of X from which the residue {2, 3, 1} are derived with respect to the moduli set {3, 4, 5}.

2.2 Moduli Selection

Moduli set form and number (choice) plays an important role in the design of any RNS system since the dynamic range, the speed and the complexity of the architecture is dependent on the choice (Deryabin, Chervyakov, & Tchernykh, 2018; Navi, Molahosseini, & Esmaeildoust, 2011)). The dynamic range which is obtained by multiplying the moduli set must be made as large as possible to avoid overflow but also small enough to reduce cost and increase speed since the speed of any arithmetic computation in RNS is dependent on the largest modulus of the set ((Abdelfattah, 2011); Baagyere, et. al, 2011;Phalguna, Kamat, & Ananda Mohan, 2018). Hence, moduli sets should be made as small as possible but with a sufficient dynamic range so as to avoid overflow (Dina & Pavel, 2013;Thabah, Sonowal, & Saha, 2018).

For computer applications it is vital to have moduli sets that ensure both efficient representation and balance(Aremu & Gbolagade, 2017; Gbolagade, 2010; N. Singh, 2016b). A balanced moduli set is one that has a very small difference between one another (Omondi & Premkumar, 2007), the traditional moduli set $\{2^n - 1, 2^n, 2^n + 1\}$ is example of such moduli sets. In cases of unbalanced moduli-sets, the largest modulus increases the cost and performance of the architecture(N I Chervyakov et al., 2020; Omondi & Premkumar, 2007; Valueva et al., 2019). A Common choice of prime modulus that does not complicate arithmetic and has good representational efficiency is $m_i = 2^i - 1$. It should be noted that not all pairs of numbers of the form $2^i - 1$ are relatively prime but can be shown that $2^j - 1$ and $2^k - 1$ are relatively prime if and only if $j \, and \, k$ are relativey prime (Omondi and Premkumar, 2007; Parhami, B., 2000). There are two (2) types of moduli set selection; the restricted and unrestricted selection.

In the unrestricted moduli set selection, prime numbers are chosen next to each other like in a series until the needed dynamic range for the computation is derived (Baagyere, 2011). The major problem with this type of moduli selection is that they do not support simple and straight forward operations in RNS. The solutions for realizing all form of mathematical operations here are based on ROMs so as to enhance the speed of execution whereas the cost of implementing ROM based RNS data converters is not cost effective (Y. E. Baagyere, 2011; Hiasat, 2019).

It has been proved and shown that power-of-2 moduli specifies any required arithmetic operation in RNS (Wang et al, 2003; Parhami, 2000; Hiasat, 2019; Phalguna et al., 2018) and thus the modulus 16 might be better than the smaller modulus 13. These power-of-2 related moduli sets form the basis for the restricted moduli set. This moduli set eliminates the need for the ROMs used in building RNS converters in the unrestricted moduli selection and uses logic gates for the realization of the basic building blocks in any RNS architecture. In reality, low-cost Moduli of the form $2^a - 1$ and $2^a + 1$ are desirable for any RNS implementation (Merrill, 1964; Parhami, 1995; Wang, et. al, 2003), hence the motivation for the restriction of moduli sets to a power of 2 and numbers of the form $2^a - 1$ and $2^a + 1$ (Parhami, 1976; Wang, et. al, 2003; Ma, 1998).

Several RNS restricted moduli sets with varying lengths have been presented in literature. The three (3) length moduli sets were reported in (Wang, et. al, (2002), Mohan, (2007), Molahosseini, (2008), Wang, et. al, (2003), Hariri, et. al, (2017); Molahosseini, et. al, (2008a; 2008b). Four (4) length moduli sets were reported in Mohan & Premkumar, (2007); Cao, et. al, (2003); Mohan, (2008); Zhang & Siy, (2008); Molahosseini, (2010) while Hiasat, (2005); Cao, et. al, (2007); Molahosseini, et. al, (2008) presented a five (5) length moduli sets in literature.

The length of moduli sets is a determinant factor in the complexity of the RNS architecture(Deryabin et al., 2018; N. Singh, 2016b). Hence, a moduli set with a four (4) or five (5) length is more complex than those based on the three (3) length moduli set (Navi, et. al, 2011). It is however preferable to choose and use a three (3) length moduli set that has a sufficiently wide dynamic range (other than the traditional moduli sets) over the 4 or 5 length (Navi, et. al, 2011).

However, irrespective of the type of moduli selection or length of moduli set adopted, moduli sets should have the following features (Baagyere, 2011; Baagyere, et. al, 2011;Deryabin et al., 2018):

i. Moduli sets should be coprime (relatively prime), that is, the greatest common divisor (GCD) (also known as the highest common factor (HCF)) between the numbers (set) must be equal to 1.
ii. Moduli sets should be made as small as possible so that the modulo operations will require a minimum computational time.
iii. The moduli set should provide straight forward computational operations using binary adder logic.
iv. The dynamic range should be large enough to avoid overflow e.g in the traditional moduli set, if n=2, M =60. Any computation above 60 causes an overflow.
v. Selected moduli set should be balanced, that is, the difference between the numbers of bits of the different moduli should be very small.

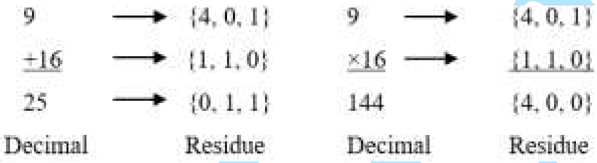
Hence, the choice of moduli set has been studied extensively (Hiasat, 2017, 2019; Pettenghi, Chaves, & Sousa, 2013). Wang, et. al, (2003) concluded that the moduli sets $2^n, 2^n + 1, 2^n - 1$ is the most efficient moduli set in terms of design of the binary/residue converters for RNS subsystems of a medium dynamic range (less than 22bits) while $\{2^{n1}, 2^{n1}+1, 2^{n1}-1, 2^{n2} \pm 1, \ldots, 2^{i} \pm 1\}$ with a length greater than 3 is the most efficient one for large dynamic ranges (equal to or larger than 22 bits).

2.3 Advantages of RNS

The field of RNS is presently gaining the attention of researchers because of the following properties;

i) RNS supports carry-free arithmetic operations: In decimal number system, when performing addition, carries propagate from the least significant bit (LSB) to the most significant bit (MSB). While carries are allowed from most significant bit to least significant bit while performing subtractions. In multiplication, partial products must be added in order to obtain the final results. In RNS, there is carry free additions, borrow free subtractions and single step additions of partial products in multiplication. The absence of carry propagation in these arithmetic operations facilitate the realization of high speed and low-power arithmetic since carry propagation is the most significant speed-limiting factor in these operations (Baagyere, 2011; Gbolagade, 2010; Omondi & Premkumar, 2007).

Example: using the moduli set {5, 3, 2} the decimal number 9 and 16 can be added and multiplied thus:



9	→	{4, 0, 1}	9	→	{4, 0, 1}
+16	→	{1, 1, 0}	×16	→	{1, 1, 0}
25	→	{0, 1, 1}	144		{4, 0, 0}
Decimal		Residue	Decimal		Residue

From the example above each column was added and multiplied modulo its base, therefore, disregarding any positional carries, arithmetic is strictly performed within each residue position. This property of RNS therefore increases computational speed since there are no carries irrespective of the size of the numbers.

ii) RNS supports parallel arithmetic operations: In RNS, digit by digit computations can be performed since there is no ordering significance between the digits. As shown in the example above, a weighted number is broken down into a set of residues called remainders and arithmetic operations such as addition, subtraction and multiplication are performed on each of the residue simultaneously or in parallel independent of one another. Hence, RNS supports parallel computations which enhance high-speed (Gbolagade, 2010; Omondi & Premkumar, 2007; Baagyere, 2011).

iii) RNS supports error detection and correction: The inherent properties of RNS suggests that a RRNS (Redundant residue number system) can be used for self-checking, error detection and correction in digital processors. Error detection and correction is usually achieved by adding one or more redundant residue digits, so any error that occurs in a single arithmetic module has a local effect and errors can easily be detected and corrected. The faulty module can be disconnected with no effect other than the reduction in dynamic range (DR). Hence, RNS supports fault tolerance (Omondi & Premkumar, 2007; Baagyere, 2011).

2.4 Disadvantages of RNS

Despite all the desirable features of RNS, its usage has not been well adopted because of the following:

i) Magnitude comparison: It was earlier stated that RNS are advantageous with respect to arithmetic operations of addition and multiplication because of parallelism. This merit cannot be extended to other useful operations and this limitation has usually constrained the widespread practical application of RNS. For example, magnitude comparison (Omondi & Premkumar, 2007). As an illustration, the representations of decimal numbers 34, 67 and 1300 in with moduli-set {7, 13, 17} produces <6, 8, 0> for 34, <4, 2, 16> for 67 and <5, 0, 8> for 1300. Unlike the situation in a positional weighted number system, the residue of 1300 is smaller than some residues for 34 and 67 which shows that positions of the digits give no helpful information. This example explains that in translation to residue representation, all magnitude information is lost.

ii) In most number systems, representation of every number should ideally be unique. This is evidently the case in a typical positional, weighted number system e.g. the ordinary decimal number system. On the other hand, with RNS it was observed that residue relative to a given modulus repeat itself after a definite period as to the residue –sets relative to a given moduli-set once the upper limit of the dynamic range has been exceeded. This implies that the number of states that can be uniquely represented is limited in RNS (Omondi & Premkumar, 2007). For example, numbers 47 and 107 with moduli-set {3, 4, 5} have the same representation which is <2, 3, 2>. The period of their RNS is 60, so the residue repeat for integer multiples of 60 e.g. the residues are the same for 47, 107, 167, 227, 287 etc. this means RNS representations generates residue that are redundant after each period (Dynamic Range).

iii) RNS conversion: RNS architecture are typically composed of three (3) main parts; a binary

- to residue converter, residue arithmetic units and a residue-to-binary converter. The overhead associated with the input and output conversions from binary to RNS and vice versa (Somayah, Mahmood, & Sorin, 2010) and the complex nature of the residue-to-binary converter architecture (Gbolagade, 2009) are major setbacks in its adoption.

iv) Residue number system also has a disadvantage in its inability to manage dynamic range overflow which unlike in the weighted number systems where overflows are efficiently handled by rounding off, truncation or saturation of arithmetic. (Taylor & Huang, 1982).

v) A complete arithmetic unit should be capable of at least addition multiplication, division, square root and comparisons. This is not so in RNS as implementing the last 3 is not an easy task (Omondi & Premkumar, 2007; Fred, 1990).

3. AREAS OF APPLICATION

Since the rediscovery of RNS, it has been well applied to areas in which critical arithmetic operations such as additions and multiplications e.g Digital Signal Processing such as digital filtering, convolution, fast fourier transform, digital image processing, Low power design, cryptography, bioinformatics etc. (Suraj, et. al, 2014; Alhassan, 2013; Dina & Pavel, 2013; Somayyeh & Amir, 2013; Baagyere, et. al, 2011; Baagyere, 2011; Gbolagade, 2010; Somayyeh, et. al, 2010; Bajard, et. al, 2009; Gbolagade, et. al, 2009; Pemmaraj, 2009; Omondi & Premkumar, 2007; Zhining & Braden, 2007; Gian, et. al, 2007; Schinianakis, et. al, 2006; Mi, 2004; Wang, et. al, 2004). This section reviews different areas and architectures where RNS has been applied.

Due to the advantages of RNS it has been majorly applied to applications needing high speed arithmetic's, fault tolerant systems, complex number arithmetic's. More recently RNS has been applied to Network routing, Cloud Storage, block chain technology and Artificial intelligence. Table 1 combines the different RNS based applications with specific areas of contributions.

Table 1: RNS Applications with specific areas of contributions

Paper	Area of Application	Challenges	Specific area of application
(Yatskiv, Sachenko, Nataliya, Bykovvy, & Segin, 2019) (Yatskiv & Tsavolyk, 2017) (T. Singh, 2014) (Campobello, Leonardi, Palazzo, & Member, 2012) (Liberato, Martinello, Gomes, Beldachi, Salas, Villaca, Ribeiro, Kondepu, et al., 2018) (Raji, Gbolagade, & Taofeek-ibrahim, 2018)	Data Communication and Networking	• Latency in routing for software defined networks • Limited and unreliable transmission in Wireless Sensor Networks • Inability for parallel execution of the distributed self-diagnosis protocol algorithm for fault detection in communication networks • Large energy consumption and unreliability in WSN	• Tabless routing in Software Define Networks • Data Transmission reliability in Wireless Sensor Networks • Fault detection in communication network • Energy saving and reliability in Wireless Sensor Network
(Vassalos & Bakalis, 2013) (Kenneth & J., 1977) (Soudris, DSgouropoulos, Tatas, & Padidis, 2003) (Pontarelli, Cardarilli, Re, & Salsano, 2008) (Luan, Chen, Ge, & Wang, 2014)	Digital Signal Processing	• increasing FIR filter efficiency	• RNS has been used to increase FIR filter efficiency • Quadratic residue number system (QRNS) was employed to accelerate complex arithmetic speed • RNS was used to build a low-cost fault-tolerant finite impulse response (FIR) filter to save logic resources.

(Wei Wang, Swamy, & Ahmad, 2004) (Toivonen, 2006) (Taleshmekaeil & Mousavi, 2010) (S. Alhassan, Gbolagade, 2013) (Nikolai I Chervyakov, Lyakhov, Nikolai, & Bogayevskiy, 2019)	Digital Image and Video Processing	• Challenge with improving speed security and low power for Digital Image Processing	• RNS scheme in digital image filtering of spatial and frequency domain to improve performance of the integration circuits
(Martinelli et al., 1990) (Abdelhamid & Koppula, 2017) (Salamat, Imani, Gupta, & Rosing, 2019) (Babenko, E, Tchernykh, & Golimblevskaia, 2020) (N I Chervyakov et al., 2020) (Samimi, Kamal, Afzalli-kusha, & Pedram, 2019) (Zhi-Gang & Mattina, 2020)	Artificial Intelligence	• To improve upon the non-linearity of neurons for optimizing the operation of Artificial neural network (ANN) • Enhancing the core multiplying and accumulating so as to improve the inference of an entire Neural Network • Complex operations and resource demanding operations in Convolutional Neural Networks (CNN) • Energy efficiency and memory bandwidth of Deep Neural Networks	• Development of a perceptron based on RNS • RNS was used to enhancing Multiply-and-Accumulate (MAC) performed during network assessment • RNSnet simplifies simple neural network operations and maps them for additional memory and data access. • Improving the efficiency of the base extension using Neural Network based on CRT and orthogonal bases. • Development of a CNN architecture using RNS to minimize the cost of the resource • RNS was used to improve the energy efficiency of DNNs and computational cores required for a high memory bandwidth. • RNS was used with the Winograd algorithm to reduce the computational complexity of convolutional neural networks (CNN)
(Mei, Gao, Guo, Zhao, & Yang, 2019) (Pandey, Mitharwal, & Karmakar, 2019)	Block Chain and IOT	• Inefficient Storage for large volume of data in block chain development. • Providing adequate security for IoT	• Applied to the optimization of storage mechanism without modifying the architecture of block chain • RNS based Elliptic Curve Cryptography cipher to provide an adequate level
(D. M. Schinianakis, Kakarountas, & Stouraitis, 2006) (Lim & Phillips, 2007) (Akinbowale N Babatunde, Jimoh, & Gbolagade, 2016) (Fournaris & Sklavos, 2016) (Kayode & Gbolagade, 2017) (Oyinloye & Gbolagade, 2018) (Akinbowale Nathaniel Babatunde, Jimoh, Oshodi, & Alabi, 2019) (I. Z. Alhassan & Ansong, 2020)	Cryptography	• Computational complexity in the total video encryption. • Attack from the side channel of fault injection in elliptic curve crypto-system. • Inefficient image scrambling image scrambling algorithm	• Point multiplication is performed using RNS circuits and data in RNS format • Use of RNS to reduce the computational complexity in the total video encryption. • Elliptic curve crypto-system based on RNS in order to mitigate attack from the side channel of fault injection. • Improve image scrambling algorithm based on three moduli set to achieve a highly reduced size of encrypted image
(Azizifard, Qermezkon, & Farshidi, 2015; Azizifard, Qermezkon, Postizadeh, & Barati, 2014)	Steganography	• Improving secured connectivity over a VoIP and 3D images • Error correction capacity and redundancy	• RNS in conjunction with DNA sequences Huffman encoding to ensure secured connectivity over a VoIP.

(Ahmadi & Salari, 2014) (Eseyin & Gbolagade, 2019) (Agbedemnab, Yellakuor, & Daabo, 2019) (Belhamra & Souidi, 2020) (E. Y. Baagyere, Agbedemnab, Qin, Daabo, & Qin, 2020)		• The decryption process of RSA public and private exponents are mostly slow.	• RNS was used improve information steganography by concealing Data within Three-Dimensional Images • Use of Redundant RNS to to hide data with limited alterations • CRT was used to speed up the slow decryption process of RSA public and private exponents • Genetic algorithm (GA) and RNS were used in order to embed encrypted text within images.
(Atta-Ur-Rahman, Naseem, Qureshi, & Muzaffar, 2011; M. Naseem, Qureshi, Muzaffar, & ur Rahman, 2016; M. T. Naseem & Muzaffar, 2012) (Priyanka, Nireesha, Kumar, Ram, & Chakravarthy, 2012) (Qureshi & Muzaffar, 2016) (Rahman et al., 2018)	Digital Watermarking	• Improving the reversible and fragile properties of watermarking • Improving the robustness of watermarking schemes	• A redundant bit is added as a watermark to some of the pixels while changing the rest into residues. • CRT is applied by selecting a two co-prime number and inverse CRT is applied to find the embedding location.
(Rajalakshmi & Nivedita, 2018) (Kehinde & Alagbe, 2018) (Mensah, Bankas, & Iddrisu, 2018)	Bioinformatics	• improving the speed of the Smith Waterman algorithm	• parallelism and carry-free propagation properties RNS was used to accelerate the algorithm on an FPGA
(Gomathisankaran, Tyagi, & Namuduri, 2011) (Kar, Sur, Basak, Sukla, & Das, 2016) (Tchernykh, Babenko, Chervyakov, & Miranda-lópez, 2018)	Cloud Storage	• collusion of independent cloud thereby reducing security.	• Homomorphic encryption scheme using RNS for cloud storage • Redundant Residue Number System (RRNS) with new method of error correction codes and secret sharing schemes.

3.1 Artificial Intelligence

Prior work applying RNS for efficient computation has largely focused on digital signal processing, cryptographic applications and general purpose ALUs. Due to its potential use in artificial intelligence and various other applications, the residue number system has become a very researched field. Some of the RNS applications are presented for artificial intelligence.

3.1.1 RNS Neural Networks

Artificial Neural Network has widely been used in various fields such as image recognition, systems modelling, forecasting, and more. An important feature that improves the operation of Artificial Neural Network (ANN) is the non-linearity of neurons (Martinelli et al., 1990). The non-linearity helps in achieving arbitrary outcome.

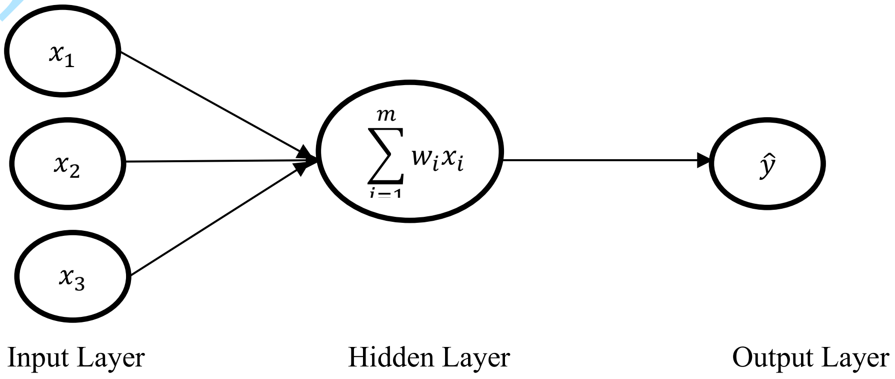


Figure 1: Basic unit of an Artificial Neural Network (Artificial Neuron)

$$z = \sum_{i=1}^{m} w_i x_i \quad \text{and } \hat{y} = f(z)$$

Martinelli, et. al, (1990) introduced a perceptron based on the Residue Number System computation. The scaling nature of RNS is used by all arithmetic operations of a perceptron in order to design a nonlinear function. Arithmetic operations are performed on the perceptron in RNS from Z, where input x_i and output are defined modM_1, the weights w_i are defined mod M_2 and Z mod $M_1M_2M_3$, where $M_1M_2M_3$ are all positive integer numbers. Figure 1 shows a basic unit of ANN, RNS was then applied in combining a few neurons in a multilayer architecture in a way to reflect the execution by a single layer perceptron. From Martinelli, et. al (1990) in equation 1, x_i are components of inputs and w_i are synaptic weights. In this technique, the learning phase indirectly replaces the modular quantities by the continuous quantities for the weights, bias and output of the adder. Mixed Radix Conversion was used for converting the function $f(z)$ from from z to $\hat{y}$, if the numbers are prime. Experimental analysis indicates that RNS perceptron has superior execution than classical perceptron. Some major drawback include the non-linear structure being non-flexible during mapping with the scaling operations, the selected moduli determines breakpoints which may limit performance. Additionally, the learning algorithm is limited to weight determination i.e. cannot be used to assess output and outcome.

3.1.2 Neural Network Inference

In 2017, Residue Number System was applied to the problem of enhancing Multiply-and-Accumulate (MAC) performed during network assessment (Abdelhamid & Koppula, 2017). A RNS based arithmetic circuit was developed to improve the core block multiplication and accumulation that would greatly improve the inference of an entire Network. An efficient modulo multiplication and addition was implemented for both element-wise using the same moduli set as the comparison, namely $\{ 2^{n_1} \pm 1, 2^{n_2} \pm 1 \}$ to fully exploit the advantages of transforming the network to the RNS system, modulo arithmetic circuits was further designed. The ReLU nonlinearity necessary for RNS end-to-end inference was then used as a comparison module for the network to prevent the overhead conversion from RNS at the end of the network. The authors modified the circuit given in (Sousa, 2007), to implement the comparison operator with various optimizations and implemented multiplication with $2^n \bmod 2^{n+1} - 1$ as a right rotate. The implementation of the RNS-based network inference was done using Bluespec System Verilog while syntheses were performed using a commercial LP65nm CMOS process and networks were implemented in Tensorflow/ Tensorpack. The framework has proven to promote the evaluation of the low power network by use of RNS inferences. RNS use, however, continues to generate overhead output and RNS multipliers options. The comparator strength can still be increased further.

3.1.3 In-Memory Neural Network Acceleration Using Residue Number System

Salamat, Imani, Gupta, & Rosing, (2019) proposed to run the neural network entirely in the memory digital domain on RNSnet which uses the RNS properties. RNSnet makes it easier to run a neural network and maps it for more memory and data access. The forward conversion logic architecture enables all input data and trained weights of the neural network to be converted to RNS. The entire neuron functionality is implemented in RNS without any backward conversion. Additionally, the scheme models the multiplication between neuron inputs and weights, by addition, subtraction and memory lookup. All parameters have been initially translated to RNS, and in this architecture there is no reverse conversion to BNS. In case of a breakdown in memory block, a range block is used to convert the value to a compatible RNS format. The final value is transferred through the activation function. Since In comparison with many known neural network implementations, the authors contrasted the efficacy of the proposed architecture. The results showed that, compared to NVIDIA GPU GTX 1080, RNSnet consumes 145:5 less energy, and achieves 35:4 pace. The results show that in comparison with state-of-the-art neural network accelerators, RNSnet can achieve an 8:5 higher power delay product.

3.1.4 Neural network method for base extension in residue number system

Babenko, E, Tchernykh, & Golimblevskaia, (2020) proposed a new way to perform base extensions using a Neural Network of a final ring. The range of base for both integer arithmetic as well as floating point arithmetic is expanded with two algorithms. The integer arithmetic approach considered CRT calculation without reducing the dynamic modulo range, which means that the converted value can have several of the dynamic modulo range. Extension method using floating point arithmetic does not uses an approximate value but uses the exact value of the conversion. The efficiency of the base extension is further improved using Neural Network based on CRT and orthogonal bases. The proposed method of base extension is characterized by the calculation of small modulo instead of calculation of the large modulo in traditional CRT. The authors implemented the algorithm using Python and considered 128-bt numbers, sets of 7 moduli was chosen. The result of this work is the definition of the base extension algorithm using Neural Network as the most suitable for practical use in various computing systems with RNS.

3.1.5 Minimizing Hardware Cost of Convolutional Neural Network

Chervyakov, et. al, (2020) proposed a new CNN architecture using RNS in the Convolutional layer of a neural network to minimize the cost of the resource needed by CNN due to the number of operations performed by CNN, which is very resource demanding (N I Chervyakov et al., 2020) .

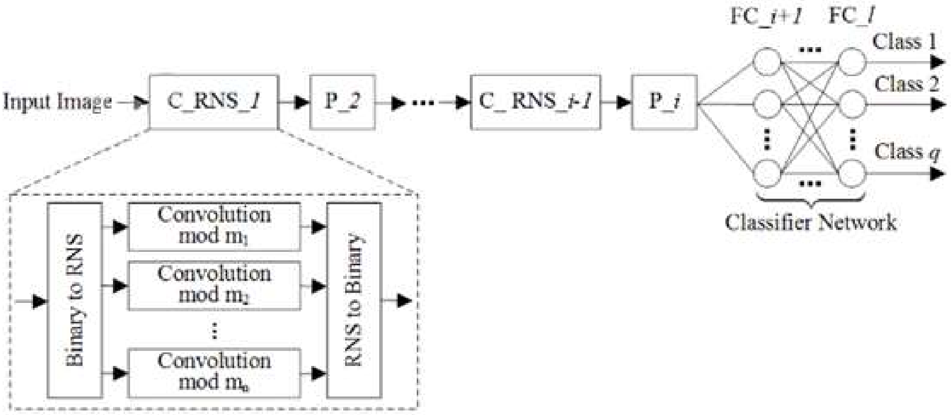


Figure 2: Architecture of CNN with RNS implementation of convolutional layers (Adapted from Chervyakov, et. al, (2020)

From Figure 2, the architecture was used in the implementation of CNN convolution layers using RNS arithmetic, containing a condensed block for converting the Binary Numeric System (BNS) to RNS numbers, the blocks for each variable is used to calculate its convolution and a block to convert RNS to BNS numbers. Reverse conversion is implemented using the new Chinese Remainder Theorem with fractions which dramatically lowers the hardware cost of implementing it. In the software implementation phase, the authors developed the CNN architecture using the RGB images of size 256 × 192 which are fed to CNN input. The first two layers are responsible for highlighting features of the image. The first layer performs a convolution using 8 filters of size 3 × 3 × 3 and stride 3. 8 feature maps of size 85 × 64 are the calculation result of the first layer are. The second layer performs the max pooling operation using a mask with a size 2 × 2 and stride 2. 8 feature maps of size 42 × 32 are the output of the second layer and are connected to the inputs of the last two layers responsible for the image classification. The third layer consists of 10 neurons, and the fourth layer consists of 8 neurons, each of which corresponds to a particular class. The hardware implementation phase focused on the comparison of RNS and two's complement. In order for maximum performance, the FPGA implementation of the CNN filters was performed as a combined circuit without memory. As was demonstrated by the authors, after all possible evaluations and simulations using Kintex7 FPGA, it was shown that the use of RNS in a neural network's convolutionary layer reduces the cost of hardware by 32.6 per cent relative to the standard binary number system approach and the use of the proposed hardware-software architecture reduces the average image recognition time by 37.06% compared to the software implementation. However, the hardware implementation method proposed is complex, new methods for implementation on FPGA would be required.

3.1.6 Res-DNN: A Residue Number System-Based DNN Accelerator Unit

Samimi, Kamal, Afzalli-kusha, & Pedram, (2019) proposed a technique based on using Residue weights and input/output data of the layers from memory to Number System (RNS) to improve the energy efficiency of Deep Neural Networks (DNN) and computational cores required for a high memory bandwidth. In RNS-based computations, a scaling technique was used to lower the bit data width to equalize the data width to the binary method. In the authors' architecture, the MAX pooling and ReLU activation function are implemented in the RNS format. The moduli-set for the RNS representation in Res-DNN is {2n − 1, 2n, 2n+1 − 1} where parameter n, the bit width of the data in this representation is n = 5. The efficiency of the architecture proposed is evaluated using the ImageNet and CIFAR-10 datasets under seven advanced DNNs. The results show that Res-DNN generates 2.5 to lower calculation energy and a binary variable.

3.1.7 Efficient Residue Number System Based Winograd Convolution

Zhi-Gang & Mattina, (2020) extended the Winograd algorithm to Residue Number System (RNS). With weights and activation represented on floating point, the Winograd algorithm can reduce the computational complexity of convolutional neural networks (CNN). A minimal complexity convolution is calculated precisely over large transformation tile (e.g. 10 × 10 to 16 × 16) of filters and activation patches using the Winograd transformation and low cost (e.g. 8-bit) arithmetic without degrading the prediction accuracy of the networks during inference. The authors implemented the algorithm in a highly optimized kernel in C on Ubuntu Linux. The program uses ILP (vector units) to boost the throughput of Winograd transforms, MRC and GEMM functions. Experimental result indicate, on average, that the scheme improved the runtime performance of 3 × 3 INT8 CNN layers by 2.02× using power efficient 8-bit arithmetic and 2.20× for 16-bit arithmetic in comparison with the standard Im2col + INT8 and INT16 GEMM baseline performances respectively measured on an Arm Cortex-A73 mobile CPU using the 8-bit quantized VGG16 model.

3.2 Information Security

3.2.1 Data Steganography

Azizifard, Qermezkon, Postizadeh, & Barati, (2014) applied RNS in conjunction with DNA sequences with arrangements made of 4 characters A, C, T and G reoccurrence and Huffman encoding to ensure secured connectivity over a VoIP. The authors used Symmetric encoding via RNS and the bit string algorithm generated was converted to a DNA sequence. Huffman string calculation allows a suitable compression level that reduces data transfer speed usage. The use of the SIP protocol to transmit hidden data as text without interference with audio packets ensures that the quality of the voice is not determined by hidden text packages. Experimental result shows that an encoded string of 24 bits is transmitted on the channel instead of a string of 75 bits which equals 53.3% of the compression of the string.

Azizifard, Qermezkon, & Farshidi, (2015) also went further developed a method combing RNS to improve information steganography by concealing Data within Three-Dimensional Images. Three Moduli Set was choosen to represent images in a 3D Space where each coordinate represents a point in three dimensional Space. Points are shown as main pixels on a three-dimensional image which are then used to determine the beginning and end of characters a line. The image is then represented as a foreground and a background image and Stego-Image is formed using alpha Factor. The authors considered alpha factor to be 0.5 For both images, background image And foreground Image are then mixed equally. Considering Alpha factor to be 0.99, background image is completely concealed in foreground image and stego-image is formed. Stego-image locate points within images using CRT technique to convert residue numbers to ASCII conversion of decimal codes to hexadecimal codes and remove key conversion to character value representation of decoded characters. Experimental investigation of the output reveals that the method is highly secure.

Ahmadi & Salari, (2014) proposed an image hiding method based on Chinese Remainder Theorem (CRT). The authors implemented the algorithm in two phases, CRT based coding is used in the first phase while the second phase involved image hiding. Four co-prime numbers (m1, m2, m3, m4) between 256 and 1024 were chosen first. The co-prime numbers represented the keys and are transmitted to the receiver end through a high- secured channel. The author implemented the algorithm to hide each pixel value of the secret image into the cover image. The authors considered a search block of 16x16 pixels in the coded cover image for each pixel of the coded secret image. The scheme was implemented in MATLAB and the results for various images with different textures, Experimental analysis shows that the scheme preserves the entire pixel values in the secret image with no changes with a higher PSNR values for the stego-images as well as a better authentication in comparison with some other methods. However, there was some distortion in the stego-image compared to the cover image.

Amine & Mamoun, (2019) introduced the idea of using Redundant Residue Number System (RRNS) codes in steganography. The redundancy and correction capabilities of RRNS codes allow the scheme to cover hidden data in certain residues in a manner that does not exceed half of the redundancy of the code with a changed number of residues. The work is based on the identification and correction capabilities of RRNS codes in the conception of the steganographic protocol to send confidential information through RRNS codes. In the RRNS schemes, all the moduli residues are coded in an equally sized binary representation of l bits for some positive integer l. Experimental result shows that the scheme is efficient compared to some the corresponding schemes while also analyzing their vulnerabilities.

Eseyin & Gbolagade, (2019) proposed a steganographic procedure to speed up RSA algorithm using Chinese Remainder Theorem for decryption of data. CRT is used to speed up the slow decryption process of RSA public and private exponents which are always chosen to be large. The slow decryption of public and private RSA exponents, often selected for their scale, is accelerated by CRT. The CRT technique on RSA algorithm is used to match data to an image resulting in a lesser probability of an attacker being able to use steganalysis to recover data. The scheme first applies encryption before hiding the image, LSB (Least Significant Bit) and RSA (Rivest Shamir Adleman) are fused together and implemented to offer extra level of protection. Experimental results obtained showed that CRT is better to deal with RSA cryptosystem complexity and substantially outperform the related state-of-art RSA cryptosystem in terms of computational cost, speed and security.

Agbedemnab, Yellakuor, & Daabo, (2019) proposed an effective scheme that can detect and repair both single and multiple errors while the redundant modulo are adequate during and after calculations and transmission. The authors used the Moduli set $\{ 2^{n-1} - 1, 2^{n} - 1, 2^{n} \}$ in other to enable the decoding of possible errors in the encoding process. CRT was applied for the reverse conversion process. The authors presented the hardware realization of the scheme for both binary to RNS and RNS to binary conversions. A theoretical analysis of the performance of the proposed scheme shows it will be a better choice for detecting and correcting computational and transmission errors to existing similar state-of-the-art schemes.

Belhamra & Souidi, (2020) proposed a distortion-less RRNS based steganographic scheme. The authors analyzed the embedding capacities and made a comparison with well-known steganographic protocols. The scheme used the redundancy and correction capacity of these codes to hide secret information in such way that the number of the altered residues does not exceed half the redundancy of the code. The first scheme embeds secrets as binary sequences, the second one operates over finite field symbols. Additionally, the third scheme embeds RNS representations of the secret data by taking advantage from their redundancy. Additionally, the authors introduced algorithms for the embedding and retrieval maps with examples and analyzed the complexity of the schemes. The scheme is as effective as the CP scheme in poor propagating conditions (T_{CP} = 0 , 8 µs) when compared with other schemes and twice more efficient in good propagation conditions (T_{CP} = 0 = 0 , 4 µs).

To insert encrypted text into images, Baagyere, Agbedemnab, Qin, Daabo & Qin (2020) suggested combining steganographic and cryptographic scheme by using genetic algorithm (GA) operators to select, crossover, and mutate RNS properties with an acceptable fusing technology. The moduli set $\{2^{n-1} - 1, 2^{n} - 1, 2^{n}\}$ was used to avoid unnecessary delays to the RNS portion, by non-derailment of the different channels. The collected residues were used for the processes of steganography and cryptography. The authors made use of Mixed radix conversion (MRC) for decoding which involves the reversal of the encoding process. The scheme was tested using MatLab® R2017b and a CORE™i7 processor and Experimental results show that only the secret message encoded in the stego-image and on the other stage, where the Stego message is further encrypted, can be deployed on the proposed scheme. In addition an analysis based on Standard Key Metrics such as visual perception and statistical steganalysis and cryptanalysis methods shows that the proposed system is stable, is not complicated with reduced runtime than the current systems due to the use of residue numbers.

3.2.2 Cryptography

With the increasing need to secure digital data. Several researchers have identified the residue number as an approach for use in cryptography because of its properties mentioned in section 1(D. Schinianakis & Stouraitis, 2016). The use of RNS arithmetic in modular arithmetic's, which reduces time complexity and thus increases the efficiency of a cryptosystem has signaled researchers to use RNS in cryptography. RNS has been applied to:

- Montgomery modular multiplication
- RSA Cryptosystem
- Elliptic Curve Cryptography

Ammar, Ai, Youssef, & Emam, (2001) proposed a technique that encodes image pixels. The image are quantized with 8 bits, using the RNS. This technique will lead to an unrecognizable image except for the user who has the key for decryption. The original digital image is read as a binary or decimal number, thereafter the prime modulo is selected that is sufficient to encrypt the digital image. The selected numbers re-converted into RNS forms thereby encrypting the image. The encrypted image is decrypted by converting each residue to its original form using look up table technique. The LUT technique is studied with the Boolean function technique, the authors found that the LUT technique is faster than the Boolean function technique. An LUT scheme is built with minimum storage requirements and used in the decoder software. C language is used to develop an encoding and decoding software programs. Experimental demonstration shows that the output is totally an unreadable image.

Lab, (2011a) established an RSA based Chinese Remainder Theorem (CRT) in order to improve RSA Cryptosystem's speed and effectiveness. Digital signature operations are divided into two to speed up operation, the digital signature operation process is given as $S = M^{d} mod\, N$ is split in two operations $S_p = M^{d_p}\, mod\, p$ and $S_q = M^{d_q}\, mod\, q$, where dp = d mod (p − 1) and dq = d mod (q − 1). Despite the acceleration achieved, RSA-CRT has proven to be especially vulnerable to hardware-fault attacks (D. Schinianakis & Stouraitis, 2016).

D. M. Schinianakis, Kakarountas, & Stouraitis, (2006) proposed an Elliptic Curve Point Multiplier (ECPM) VLSI-RNS architecture whose efficiency determines mainly operation of the entire cryptosystem.

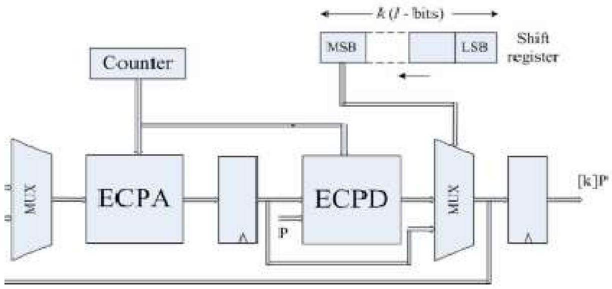


Figure 3: Architecture proposed for ECPM , Adapted from (D. M. Schinianakis et al., 2006)

The field circuit is replaced with RNS by selecting an adequate RNS dynamic range. Point multiplication is then performed using RNS circuits and data in RNS format. The authors designed a modulo based circuits in order to synthesize RNS adders, subtracters, and multipliers. A multiplexer is placed at the inputs of each RNS structure, in order to choose the operands that will be processed while a decoder is placed at the output of each RNS structure, which drives the result to an appropriate register. ECPM was synthesized in a Xilinx Virtex 2–Pro (XCVP125, FF1696) Experimental findings suggest that a scalar point multiplication was rendered with an FPGA by approximately 82 per cent improvement in the execution time, but its implementation uses the large range of RNS chosen in comparison with other implementation.

Bajard, Duquesne, Ercegovac, & Meloni, (2006) analyzed the complexity of Residue Number Systems (RNS) for primary field (GF(p)) and Trinomial Residue Arithmetic for binary field (GF(2k)) in the summations of products. One of the best domain of application is the Elliptic Curves Cryptography where addition and doubling point's formulas are composed of products summation.

Lim & Phillips, (2007) presented a new residue number system implementation of the RSA cryptosystem. The RSA public key cryptosystem is implemented on a low power microprocessor using modular arithmetic. Two versions of RSA were built on Tensilica's Xtensa processor to evaluate the approach. Montgomery's algorithm which reduces an integer product AB modulo the integer N is use for both multiprecision and RNS versions for the reduction step. The authors explored and developed a hybrid version of the Montgomery's algorithm. The RNS implementation was compared against a baseline implementation that uses non-RNS multi-precision methods, the new RNS implementation executes in 67.7% fewer clock cycles. The hardware support requires 42.7% more gates than the base processor core.

Akinbowale N Babatunde, Jimoh, & Gbolagade, (2016) proposed a scheme that uses RNS to reduce the computational complexity in the total video encryption. The proposed scheme which will be implemented using Java programming language is envisaged to efficiently secure video data from unauthorized access during transmission and storage. MPEG IV compression algorithm is used on the on any video data format to compress the video data such that the size of the video file can be reduced and the compression ratio computed. The compressed video file is then separated into frames after which the pixel value of each frame is computed. RNS moduli set is then applied on the pixel values to obtain the residue of each frame which is then saved on a lookup table. RNS column addition is applied to the output of the first encryption stage followed by a RNS row addition. Scaling is then performed on the output file to further reduce the video size.

Fournaris & Sklavos, (2016) suggested an elliptic curve crypto-system based on RNS in order to mitigate attack from the side channel of fault injection, The authors presented an algorithm using Leak resilience Arithmetic (LRA) and RNS as an add-on for Power Analysis (PA) and Fault Injection (FA) counter measurement which does not use a redundant RNS modulo as an FA countermeasure as opposed to previous RNS proposals. A well designed 'initial permuter technology' is used as an effective PA countermeasure. Test analysis appears to be able to offer high PA-FA resistance when combining traditional PA-FA countermeasures with light RNS countermeasures. The suggested solution does not involve C-safe and sign shift failure attacks and is not using Non adjacent scalar (NAF) type.

Kayode & Gbolagade, (2017) further developed an efficient decryption algorithm using CRT and strong prime to speed up the modular arithmetic in traditional RSA implementations. p-1, p+1,q-1 and q+1 are factored to get their prime factors and the respective moduli. The modular exponentiations is computed with prime factors of p – 1 as the modulus. CRT is then applied to generate y1 = cd mod (p – 1). Thereafter, CRT is applied to generate the plaintext M = cd mod n based on X1 and X2. In the scheme the decryption key holder performs the decryption procedure; C^d mod n. The encrypted message can be recovered by repeated modular exponentiation. Primitive traditional method, Chinese Remainder Theorem method and Chinese Remainder Theorem and strong prime criterion were used for comparisons. Experimental cost analysis shows that the result takes about 37% of the computational cost, which is almost 3.2 times faster than the Chinese Remainder Theorem based method.

Oyinloye & Gbolagade, (2018) implemented an improved image scrambling algorithm based on three moduli set to achieve a highly reduced size of the encrypted with an enhanced secured image. Pixel Scrambling is performed by random number generation and an additional layer encryption with Residue Number System (RNS) Forward Conversion with respect to the moduli set {2n-1, 2n, 2n +1} for the encryption stage. Reverse conversion with the moduli set {2n-1, 2n, 2n +1} is used set to decode the pixel while pixel unscrambling is done by the saved sequence from the random number generator. The image is first encrypted by moving the position of each pixel in the image without having to change the pixel value of the image. The original image is read pixel by pixel, row by row and each pixel will assume a new position in the scrambled image. The original image is read pixel by pixel, row by row and each pixel will assume a new position in the scrambled image. The new position is chosen using random number generation from the random number generator. The key will be generated as a matrix during the encryption process and also the key saves the position of each pixel in the encrypted/scrambled image. The encryption layer transforms the scrambled image to moduli images which automatically adds an extra security layer to our data (Image). Experimental results shows an enhanced image encryption process and a more efficient decryption process without loses of any important information of the recovered plain image and approximately 92% compression ratio of the decrypted image. It produces cipher images that is at least 2/3 the size of the original image. This saves a minimum 25% of internal memory allocation.

Akinbowale Nathaniel Babatunde, (2019) presented a methodology for an image cryptosystem to tackle the problems of many existing digital image cryptosystems. Gray code number system design is used to transform the pixel information of an image. The images are first binarized and based on the location of a particular pixel value an algorithm is selected for its transformation. The gray code is later transformed and reconstructed for the image. The second phase is used to encrypt the pixel values of m by n images such that two neighboring pixel values with the same RGB information will not have the same cipher information. The encryption scheme consists of four steps, i.e. image conversion in binary codes, algorithm selection based on pixel position, gray-code transformation and reconstruction of RBG information. Experimental analysis from the encryption and decryption designs suggests that the proposed technique will efficiently encrypt digital images.

I. Z. Alhassan & Ansong, (2020) proposed a cryptosystem that uses a modified k-shuffling technique to scramble pixels of images and further decomposes them using Residue Number System. Simulations were performed using two moduli sets with the modified k-shuffle technique. Among the chosen moduli sets, the even moduli set optimizes and completes execution using less time as compared to the traditional moduli set. The scheme also showed resistance to statistical attacks (histogram, ciphertext, correlation attacks) and a significant reduction in the size of cipher images which enhances the speed of transmission over network. The authors modified the traditional k-shuffle algorithm so as to enable image shuffling. MATLAB Simulink simulation tool was used for simulations and interpretations. Result analyzes show that both simulations can secure images without any loss of information and also depends upon the module set for the time taken for a full encryption/decryption process. Due to the smaller pixel values and size with high statistical strength resistances (brute forces, correlation coefficient, entropy and histogram information), and a plain-image recovery with a small or no loss of any inherent function, image can be transmitted more fast over the networks.

3.2.3 Digital Watermarking

Across the fields of copyright and digital asset protection, RNS has been widely applied. The Chinese Remainder theorem has been used to enhance the watermarking of digital images for the incorporation of watermarks using the discrete cosine transform domain (Patra, Kishore, & Bornand, 2011). The primary goal of CRT watermarking is to achieve a sense of imperceptibility and to resist multiple image transformation attacks, although it remains a challenge to reduce low Tamper Assessment Feature (TAF).

Atta-Ur-Rahman, Naseem, Qureshi, & Muzaffar, (2011) introduced a reversible and fragile watermarking scheme using the Residue Number System (RNS). A redundant bit is added as a watermark to some of the pixels while changing the rest into residues. An extra bit is added to make the watermarked pixels to become nine bits and the residues also becomes nine bits making the medical image more secure by confusing the attacker about the location of the embedded watermark. Even parity of residues were calculated and added to the corresponding residues. At the decoding stage, a parity check is performed on each residue and if each residue has same parity as in encoding side then the RNS algorithm is applied on residues to get back pixel for each watermarked pixel. MATLAB 7.0 was used for experimental purpose. Experimental result shows that the watermark information is not sent on the transmission channel thereby, providing maximum security, however, the original host image is altered to provide security.

Priyanka, Nireesha, Kumar, Ram, & Chakravarthy, (2012) proposed Chinese Remainder Theorem (CRT) and Aryabhata Remainder Theorem (ART) based watermarking scheme that work in the Discrete Cosine Transform (DCT) domain. The embedding procedure involved the application of DCT conversion to the selected 8 x 8 block and a watermark bit is selected randomly. CRT is applied by selecting a two co-prime number and inverse CRT is applied to find the embedding location. The extraction procedure is the same embedding procedure except for the condition of determining the absolute difference. After which, the value of diff is compared with the absolute difference D. Experimental results have shown that the scheme makes the watermark perceptually invisible and has better robustness to common image manipulation techniques. ART-based algorithm can be applied to any kind of moduli and its computation cost is less than that of the CRT-based algorithm. The scheme is able to withstand the JPEG compression. The PSNR value of the proposed scheme is found to be higher.

Qureshi & Muzaffar, (2016) introduced a watermarking scheme which is robust by using RNS and chaotic algorithms to keep the image fragile. Pseudo-random signal is added to the image that is below the threshold of human perception. Region of Interest (ROI) is extracted from the original image and RNS properties is applied to the selected regions. Chaotic sequence of a particular length is generated. Each bit is spread by a large factor chip-rate to obtain the spread sequence while the spread bit is modulated by a binary pseudo-random noise sequence. The result is amplified after performing modulation with a scaling parameter. The spreaded watermark is then embedded in Region of None Interest (RONI) pixels based on chaotic key generated. RONI pixels are rearranged making the watermark exist in both the RONI and ROI is residued. For extraction the ROI pixels with RONI pixels to get the original image after computing the hash and separating the ROI from the RONI watermarked image. The scheme is blind and experimental analysis shows that the method surpasses any of the previous techniques, although computationally difficult, because they require so many procedures and algorithms to protect them.

P. Singh, (2016) proposed a watermarking technique for digital gray scale images that is based on utilizing congruence's in number theory and its generalizations in abstract algebra which have been developed in the context of image compression. The watermarking scheme is based on CRT with a double Density Discrete Wavelet Transform framework individually which allows for the possibility of directly watermarking the Image bit- stream. The watermark region is selected based on the transformation level. Initially, the cover image and watermark are transformed into spatial domain using Double Density Wavelet Transform and then singular values of these transformed images are combined using CRT coefficients. Experimental results was provided in terms of Peak signal to noise ratio (PSNR), Mean Squared Error (MSE), Correlation and Weighted Peak signal to noise ratio (WPSNR) to demonstrate the effectiveness of the algorithm. The embedded watermarks cause imperceptible distortion at levels that provide reliable detection. The authors also provided a comparison and validation of the methodology in terms of PSNR and MSE in order to prove the CRT based DD-DWT approach is better as compared to traditional DWT-CRT based approach. The scheme PSNRs of the watermarked images are always greater than 40 dB and it can effectively resist common image processing attacks, especially by JPEG compression and different kind of noises (Gaussian noise, speckle noise, salt & pepper noise, Poisson noise etc.). An option for selecting a wavelet from different type of wavelets was also provided, while experimental result showed that Symmlet and Mexican hat wavelets proved better as compare to traditional wavelets like Haar and Daubechies.

Bhangale, Raje, Maurya, & Gawad, (2017) proposed the combination of an advanced encryption standard (AES), hybrid of DWT and DCT watermarking techniques and residue number system (RNS) for image security. A grayscale image is used as input to AES-128 using a key, the output is watermarked using hybrid discrete wavelet transform and discrete cosine transform. DWT technique is applied to segment the cover host image so that it gives four non- overlapping sub-bands, the sub-band LL1 is then divided into 8 x 8 blocks. Thereafter DCT watermarking is applied to each block in the chosen sub-band. A gray-scale image having pixels equal to the number of 8 x 8 blocks is reformulated into a vector of zeros and ones. Two uncorrelated pseudorandom sequences are then generated and embedded in the entire pseudorandom sequence as either W0 or W1, with a gain factor, while reverse process is carried out for retrieval of original image. The retrieved and original image are compared on the basis of PSNR and MSE values. The entire process is designed in Java. Experimental result for the PSNR and MSE values for the proposed combination are 36.3934 dB and 14.9188 respectively. Hence, the authors concluded that proposed combination is highly efficient as it enhances image security and provides authentication.

Rahman et al., (2018) introduced a secure spatial domain and hybrid watermarking technique for obtaining watermark and improving the robustness and fragility of the host medical image using product codes, chaos theory, and residue number system (RNS). The host medical image is separated into two parts, namely, the region of interest (ROI) and region of noninterest (RONI) using a rectangular region. The RONI part is used to embed the watermark information. Additionally, two watermarks are used: one to achieve authenticity of image and the other to achieve the robustness against both incidental and malicious attacks. Experiment was conducted in MATLAB, the test image is ultra- sound image of size 194× 259 greyscale pixels, and the watermark logo of size 30 × 30 was considered. The scheme is highly fragile and unrecoverable in terms of the host image, but it is significantly robust and recoverable in terms of the watermark. The authors showed the effectiveness in terms of security, robustness, and fragility of the scheme by simulations and comparison with the other state-of-the-art techniques. The scheme is robust against a variety of attacks because of the error correcting codes utilized.

3.3 Digital Signal Processing

3.3.1 RNS-Based Finite Impulse Response and Infinite Impulse Response (IIR)

In digital signal processing applications, Residue Number System (RNS) has been commonly utilized over Finite Impulses Response (FIR), RNS has been used to increase FIR filter efficiency (Chang, 2015). The FIR-filter architecture based on RNS includes for each parallel channel direct transform and module summation and the opposite transformation. The main transformation from a Two-Complement System (TCS) into the RNS is to find the residue of each module (Del Re, Nannarelli, & Re, 2002)(G C Cardarilli et al., 2003).

Kenneth & J., (1977) presented a technique for implementing a finite impulse response (FIR) digital filter in a residue number system (RNS). A new hardware implementation of the Chinese Remainder Theorem. The authors suggested an effective translation into natural numbers of residue coded outputs. The scheme showed a numeric example of residue encoding, residue arithmetics, and residue decoding principles for FIR filters. A 64th-order, dual-band pass filter RNS implementation can be compared to several alternative filter structures to show the differences between speed and hardware complexity. The addition and multiplication of residues were implemented by high speed bipolar PROM-stored lookup tables, which resulted in delays of 30 ns each time. The RNS implementation was compared to conventional FIR structures with multipliers of shift-add and array and with a little slice in which linear combinations of the filter coefficients are saved in ROM. Although cost-speed sacrifices were apparent, the layout of the RNS seems favorable to compete with the traditional filters. For low-order FIRS, the architecture of bit pieces is faster and cheaper given fixed filter coefficients are permitted with the application.

Fred, (1990) presented a DFT implementation which is based on the fusion of three namely number theoretic transforms, modular arithmetic and distributed arithmetic. Quadratic residue number system (QRNS) was employed to accelerate complex arithmetic speed. The authors reduced the complex multiplier budget to its lowest bound. The authors investigated the finite impulse response (FIR) structures and the prime factor transform (PFT) was shown to possess many desirable attributes. In the overflow management problem introduced with the QRNS, a finite impulse response form of the DFT, known as primary factor transformation (PFT) was used. In order to implement the PFT and the required QRNS overflow scaling units, a fast and compact distributed arithmetic filter (DAF) and number system converter is designed. The authors showed that the best architecture for PFT and QRNS integration is the distributed arithmetic filter (DAF). The basic DAF engine is shown to apply directly to the magnitude scale problem needed to avoid overflow of the dynamic range. The resulting design would include just a few generic QRNS-DAF engines, and the internal architecture and data flow would be simple and standard VLSI. Experimental results show that the resulting machine possesses megahertz class output in a restricted hardware range of 16- to 24-b data.

Soudris, DSgouropoulos, Tatas, & Padidis, (2003) extended the RNS-based filter to define a Computer Aided Tool (CAD) tool development methodology in order to automatically generate a usable VHDL description of each of the RNS designs. Each channel was used to implement a FIR modulo mj, where inputs and coefficients are in their residue representation. The preprocessing stage decides addition and products and Bit Reduction stage uses a modulo arithmetic property to minimize the output word length of the first stage. The final stage of mapping maps its residual modulo in the performance of the second stage. In contrast to similar RNS filters, conventional binary filters have been introduced and results are presented. The authors also implemented a RNS Full Adder-based FIR, Scaling, Converter, Multiplication and Accumulator Units of a DSP Architectures. C++ was used to build a CAD tool which implements the FA-based RNS which automates the design procedure, starting from the function expression and ending up with the VHDL description.

Gian Carlo Cardarilli, Nannarelli, & Re, (2007) proposed the use of the RNS number representation as a method for power reduction in DSP architecture implementation. Both Two Complement System (TCS) and the Residue Number System (RNS) representations were used to implement each filter. For evaluating results, a model connecting power consumption with complexity circuit (area) and the position of interconnections is used. Research was carried out on the implementation of both ASIC-SC and FPGA and various inputs were analyzed with the use of region (A), power (P) and global index ratios (GI). Experimental findings show that RNS representation can be reduced considerably. The scheme reduced complexity and reduced the connectivity power. Both the complexity reduction and the position of the RNS representation have been used in FPGA implementation.

Pontarelli, Cardarilli, Re, & Salsano, (2008) Proposed the development and study of RRNS-based FIR complete fault-tolerant filters using fault concealing to provide an RNS reverse converter with an optional defect tolerance. For each module the authors use CRT and the reverse conversion method error correction capabilities are retained. VHDL is used to define the filters in Xilinx Virtex V FPGA when implementing h. In order to secure blocks that perform final RNS-based FIR calculations, the authors avoided using a trivial modular redundancy. For the CRT blocks and 3 legit blocks, LUTs are used. Experimental results show that the use for the implementation of the TMR of the error correction blocks allows savings of over 33 percent of the resource.

Luan, Chen, Ge, & Wang, (2014) Introducing a FIR filter to save logical property for low-cost fault-tolerant finite impulses. The scheme is based on the redundant residue number systems (RRNS) by removing soft errors in space applications caused by single event upset (SEU), whereby only one set of three modulo residues is required in a binary converter based on the CRT. When a soft error happens, only one small-sized first-in–first-out and rollback operations are needed to refresh the FIR filter corrupted by SEU. The authors used a structure that includes four L-tap modulo mi(i = 1,2, 3,4) FIR filter branches, one three-moduli set RTBC adopting CRT theory, one comparison module denoted as COMP and one first-in–first-out (FIFO) denoted as input-FIFO. Four L-tap FIR filters have the same structure implemented in transposed form with different modulo. Additionally, the Hamming code is adopted to protect the storage information. The data are then read from input-FIFO and fed to the four L-tap FIR filters after modulo operation. The outputs of the first three filters (mod m1, m2 and m3) are then used to recover the final output data y by the RTBC and the fourth filter (mod m4) is redundant. The authors went further to develop a low-cost fault-tolerant FIR filter based on RRNS and a three-moduli set RTBC is developed, which achieves 21, 27 and 31% of area reduction compared with the standard RRNS.. Theoretical analysis and fault injections are performed to validate that there is no fault missing event. Experimental analysis shows that the scheme can save 21% cell area compared with the conventional RRNS method.

3.3.2 Digital Image and Video Processing

Digital image processing consists of several parts ranging from digital image filtering, digital image enhancement to image transformation and picture compression. Over the years, researchers have used Residue Number Systems in digital image processing methods to increase speed and low power. (Taleshmekaeil & Mousavi, 2010; Wei Wang et al., 2004). This section describes major applications of RNS to the implementation of digital image algorithms.

Wei Wang et al., (2004) proposed a RNS image coding scheme that offers high-speed and low-power VLSI implementation for secure image processing. The entire image is encrypted and does not require any additional component other than a standard RNS system. CRT is modified and its associated residue-to-binary conversion and moduli selection methods are used for encryption. The design of an encoder and decoder pair for the greyscale image is carried out using MATLAB tool and some VLSI tools. An efficient low-cost moduli set is used image and the image processing. The small-wordlength parallel outputs of these RNS image processors are arranged into an encrypted bitstream by a certain order. The decoder is a Residue to Binary (R/B) converter to recover the encrypted bitstream in the corresponding order back to the processed image data. Then, all the building blocks and the architectures of the encoder and decoder from the Matlab are coded in the VHDL language. Next, the codes are executed at the register transfer level (RTL) to verify the correctness of the designs. The logic synthesis is carried out to optimize the designs, and the gate-level simulation performed. Finally, the placement and routing are carried out automatically to generate the layout. The preliminary results of the Matlab simulation demonstrate the security ability of the proposed image coding scheme. A performance evaluation in terms of the area, delay and power consumption is carried out at the layout level.

Toivonen, (2006) investigated image and video convolutions based on Fermat number transform (FNT). An arbitrary integer is used which gives much wider variety in possible moduli, at the cost of decreased transform length of 16 or 32 points for QP. Secondly, residue number system (RNS) is used to enlarge the effective modulus, while performing actual number theoretic transforms with smaller moduli. The authors designed an efficient reconstruction circuit based on mixed radix conversion for converting the result from diminished-1 RNS into normal binary code. The circuit is implemented in VHDL and found to be very small in area. The circuit for converting a number from the RNS modulo and into Normal Binary Code (NBC) was implemented in VHDL hard-circuit for converting a number from the RNS modulo and into NBC was implemented in VHDL hard-ware description language and synthesized for UMC 0.18-m CMOS VLSI standard cell technology using Synopsys Design Compiler version 2000.05-1. The size of the resulting circuit was very small, 261 gates. Experimental result shows that a particularly useful RNS is obtained with moduli and which has the dynamic range of about 24 bits, the authors went further to develop an efficient reconstruction circuit for this case, based on the mixed radix conversion.

Taleshmekaeil & Mousavi, (2010) proposed a scheme in digital image filtering of spatial and frequency domain based on the Residue Number System which leads to performance of the integration circuits with high speeds and high security and low power for Digital Image Processing. The selected mask in the residue system is moved on the desired image and gray levels of the pixels of new images are computed based on mask values, all computations of this scheme are carried out in the Residue Number System. Circuits of binary to residue conversion are designed while Matlab Software is used with VLSI tools for simulation. Binary systems' conversions are designed to residue system (B/R) and vice versa Residue to Binary (R/B). The scheme is simulated and compared with previous methods using Matlab simulation tool show ability of suggested design for filtering of given images. Results show that the proposed scheme has the ability for digital image filtering.

S. Alhassan & Gbolagade, (2013) proposed a security enhancement scheme for digital images. Residue Number System (RNS) to Decimal (R/D) encoding and decoding using the moduli set and a modified Arnold transform algorithm are used to achieve the enhancement. RNS to Decimal (D/R) is used for encryption to decompose a plain image into three residual images. The residual images are fused together and encrypted using the modified Arnold transform. In the decryption process, modified Arnold transform is used to decrypt the cipher image which is then decomposed into three residual images. An R/D converter (decoder) is then used to recover the plain image. The scheme is simulated on digital images of different sizes using MATLAB. Results obtained show that the scheme can effectively encrypt and decrypt images without loss of any inherent information. The scheme also offers firm resistance to statistical attacks such as histogram, brute-force, correlation coefficient and key sensitivity.

Nikolai I Chervyakov, Lyakhov, Nikolai, & Bogayevskiy, (2019) proposed an RNS-based imaging smoothing method. The principal idea of this approach is to replace the complex divisions operation in the RNS, multiply by a factor of certain value all fractional numbers and complete them. All subsequent calculations are carried out only by the integer numbers, because of these actions, the computing error that results from rounding does not significantly affect the results of the image filtering.

3.4 Digital Communications and Networking

3.4.1 Residue Number System Based Multicarrier CDMA for Broadband Mobile Communication Systems

Madhukumar, Chin, & Premkumar, (2000) proposed a method to *enhance* the bandwidth efficiency *of* an MC-CDMA system by using a residue number based representation for information *symbols.* Residues are mapped into a *set* of orthogonal sequences and *are* transmitted in parallel. A multicarrier modulation scheme are used for both transmission and reception of residue *channel.* Appropriate error correction methods *are* employed for enhancing the performance *the system.* A binary to residue converter converts the coded bits to residue values and are subsequently interleaved. The orthogonal code sequence corresponding to each residue value is selected and spread in frequency domain. In the receiver side, parallel residue values corresponding to a symbol are reconstructed after demodulation and dispreading. The residue channels are de-interleaved and then converted back to binary representation after the error correction using inverse residue transform. Extensive simulation for different channel parameters in a multipath environment with an assumption that the system has a constant chip rate of 4.096 Mcps. The performance of the proposed system is analyzed in a slow fading Rayleigh channel. Considering the bandwidth efficiency, and robustness against channel impairments, the scheme can be considered as an alternative to high-speed data transmission and even as a candidate for next generations of mobile communication systems. However, the outer coding and outer interleaving modules common to all communication systems are not discussed.

Shahana, Jose, Jacob, & Sasi, (2008) proposed a concatenated coding scheme that improved the performance of Orthogonal frequency division multiplexing (OFDM) based wireless communications. Redundant Residue Number System (RRNS) code as the outer code and a convolutional code as the inner code. A direct conversion of analog signal to residue domain is done to reduce the conversion complexity using sigma-delta based parallel analog-to-residue converter. The bit error rate (BER) performances of the system under different channel conditions was investigated. These include the effect of additive white Gaussian noise (AWGN), multipath delay spread, peak power clipping and frame start synchronization error. The simulation results show that the proposed RRNS-Convolutional concatenated coding (RCCC) scheme provides significant improvement in the system performance by exploiting the inherent properties of RRNS. The conversion of analog signal to residue domain is done using a sigma-delta based parallel analog-to-residue converter. Comparing the received redundant residue digit with the concatenated coding (RCCC) scheme offers significant improvement in BER performance under different channel conditions. The A/R converter is simulated for 8-bit resolution using the moduli set as (5, 7, 8, 9, 11) to avoid overflow. The modulator used is 2-2 cascaded MASH architecture with OSR of 16 and single-bit quantizer using MATLAB® Simulink models. This coding scheme offers significant improvement in BER performance for the OFDM system. The performance analysis shows that the proposed RCCC scheme is suitable for OFDM as it improves the-3 10 tolerance of system to channel noise, multipath effects, timing errors and peak power clipping.

3.4.2 RNS-Based Software Defined Networking (SDN)

Software Defined Networking (SDN) is a networking method that uses software-based controllers and Application Programming Interface (APIs) to handle network traffic and communicate with the external hardware components rather than a conventional hardware system such as routers and switches (Kreutz et al., 2015).

Open Flow is a popular SDN protocol. Although, an Open Flow enabled switching network can hardly fulfill the main network requirements in reactive mode, given that its hardware flow table implementation on a multi-vendor core network is severely restricted to the total number of end-to-end active flows on that path. KeyFlow was used for developing simplified core network element for packet forwarding, as was demonstrated by Martinello et al. (Martinello, Ribeiro, Oliveira, & Vitoi, 2014) using RNS to create an opt-in option to OpenFlow. The output ports of the core interlockers used by the route can be identified by the residues of the large integer route path ID using the core network switches ID of a route represented by the RNS module. This enables a highly scalable KeyFlow fabric to overcome the table lookup bottleneck in the data plane of the SDN core network.

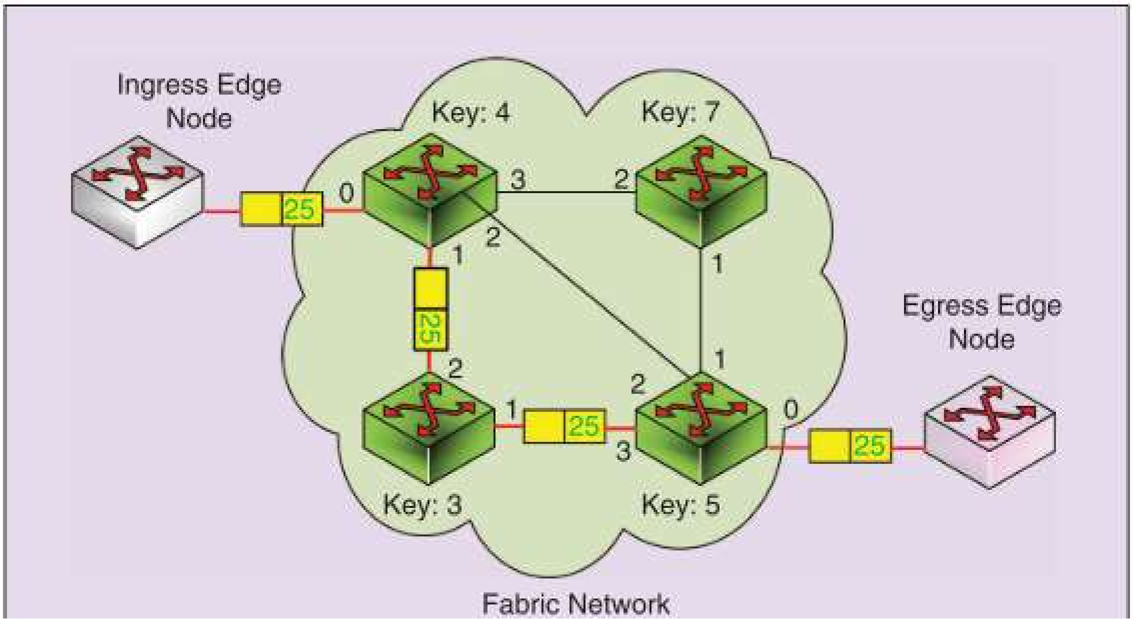


Figure 4: RNS-Based Network Switching Strategy (Keyflow) - Adapted from Martinello et al. (Martinello et al., 2014)

KeyFlow reduced round trip time by more than 50 per cent, especially for flow table networks. It also reduced the active flow status of the network by more than 30 percent.

Gomes, Liberato, Dominicini, Ribeiro, & Martinello, (2016) introduced a new intra-domain resilient routing system in which edge-nodes set a route ID to select any existing route as an alternative to safely forward packets to their destination. In KAR routing system, a route is defined as the remainder of the division between a route ID and a set of switch IDs along the path(s) between a pair of nodes. KAR-enabled switches explore the existing routes by using special properties of Residue Number System as our encoding technique. Packets are deviated from the faulty link (liveness condition) with routing deflections. Deflected packets are guided to their original destination due to resilient forwarding paths added to the route ID. Three deflection methods are discussed along emulation experiments. Results show that KAR efficiently allows deflected packets to automatically reach their destination, imposing a bound on packets disordering measured in TCP throughput. Our proposal combines a source routing technique based on the Residue Number System (RNS) with Driven Deflections property in order to enable efficient loop-free routing even in the event of a link failure. In summary, KAR poses as a fast failure reaction scheme enable high forwarding performance with the use of simple, low-cost switches and resilience, as the protection paths enable the packet delivery after a failure through loop-free alternative paths without any reconfiguration on the network nodes.

Liberato et al., (2018) introduced Residue Defined Networking Architecture (RDNA) as a new approach for enabling key features like ultra-reliable and low-latency communication in MDC networks. RDNA explores the programmability of Residues Number System (RNS) as a fundamental concept to define a minimalist forwarding model for core nodes. Instead of forwarding packets based on classical table lookup operations, core nodes are tableless switches that forward packets using merely remainder of the division (modulo) operations. By solving a residue congruence system representing a network topology, we found out the algorithms and their mathematical properties to design RDNA's routing system that (i) supports unicast and multicast communication, (ii) provides resilient routes with protection for the entire route, and (iii) is scalable for 2-tier Clos topologies. Experimental implementations on Mininet and NetFPGA SUME show that RDNA achieves 600 ns switching latency per hop with virtually no jitter at core nodes and sub- millisecond failure recovery time. An extensive scalability analysis investigated 23 types of 2-tier Clos network benchmarking RDNA with existing approaches. Our evaluation quantifies the RDNA encoding for unicast, resilient routing and multicast communication, considering micro datacenters topologies as a reference design. Experimental results show that RDNA achieves low-latency (600 ns) in the core at 10 Gbps and virtually no jitter.

Valentim et al., (2019) proposed a RDNA Balance that exploits elephant flow isolation and source routing in core nodes. Flow classification operations are performed on the edge using features of the OpenFlow protocol. The results show that with this approach it is possible to provide a simple, scalable, and programmable load balancing for data centers. This work elaborates a solution for the load balancing prob- lem in DCNs, named RDNA Balance. We leverage the SSR mechanism used in the RDNA architecture, based on fabric core nodes and programmable edge nodes, to perform load balancing at the source in a reactive and centralized way. The mechanism aims to isolate elephant flows from mice flows to improve the bandwidth available to the elephant flows while decreasing the latency experienced by mice flows. The RDNA architecture is composed by the three elements represented in the Figure 1(a): i) RDNA Controller, a logically centralized controller used to configure polices and manage the switches; ii) edge switches, which insert route identifiers in the packets specifying a path to the flow; and iii) core switches that forward packets based on a modulo operation (remainder of division) between the route identifier and the switch identifier. The results show that the mechanism proposed by RDNA Balance is able to migrate routes with low data loss rate, without compromising the communication between servers. Besides, the results show the mechanism offers flexibility for path selection, since the migration is simple and manageable.

3.4.3 RNS-Based Wireless Sensor Network (WSN)

Junior, Fernando, Nascimento, Carlos, & Albini, (2011) The technique proposed in, using the modularity nature of residual code, is also an application of RNS in network systems that reduces the number of message drop-offs caused by malicious nodes, buffer overflows, node movement and collisions in ad hoc networks.

Yatskiv & Tsavolyk, (2017) proposed an enhanced special module system for correcting codes based on Residue Number System. The research of algorithms hardware and time complexity for error detection and correction has been conducted on the basis of the Residue Number System correcting codes, using the proposed module system. Implementation of error detection and correction devices on the Field-Programmable Gate Array, using a special module system, provides reduction of hardware costs approximately by 26% with the projection method, that is about 6.5 thousand logic elements, and approximately by 10% with the method of the syndrome calculation. Conducted research showed that the implementation of error detection and correction devices on programmable logic integrated circuits with the use of special module system provides reduction of hardware costs approximately by 26% with the projection method, that is about 6.5 thousand logic elements, and approximately by 10% with the method of the syndrome calculation. The use of special module system improves also the decoder efficiency approximately by 18% with the projection calculation method and by 9% with the method of the syndrome calculation.

Raji, Gbolagade, & Taofeek-ibrahim, (2018) proposed RNS has also be used to enhance the reliability of WSN by reducing the mean energy consumption of each sensor node. Authors in proposed approach relies on a packet-splitting algorithm based on the Chinese Remainder Theorem CRT) and is characterized by a simple modular division between integers. An analytical model for estimating the energy efficiency of the scheme is presented, and several practical issues such as the effect of unreliable channels, topology changes, and MAC overhead are discussed. In addition, RNS is used to improve WSN reliability by reducing each sensor node's average energy consumption. The approach is based on the Chinese Remainder Theorem and is characterized by a simple modular division among integers using packet dividing algorithms. An empirical model for estimating the scheme's energy efficiency and the impact of unstable channels, changes to topology and the overall MAC still remains a challenge. The method does not tolerate collision and retransmission effects at the MAC layer, however there is significant power saving performance, flexibility, and a reasonable energy distribution at all nodes.

3.5 Bioinformatics

RNS has been used to parallelize the computational demanding Smith Waterman Algorithm (SWA) that is used for the search for biological sequence databases in Bioinformatics.

Kehinde & Alagbe, (2018) suggested the architecture of the RNS based Smith Waterman Algorithm for a moduli set { $2^{n-1}-1, 2^n-1, 2^n$ }.The authors presented the hardware realization of the proposed RNS with a binary to residue converter, an RNS processor and a Residue to Binary Converter. The challenge for implementing SWA-based RNS is how to effectively manage memory using an efficient algorithm and how to effectively explore RNS parallelism on a multi-core CPU or GPU.

Rajalakshmi & Nivedita, (2018) proposed the design and implementation of Smith–Waterman algorithm. The aim of this work is to improve the speed of the algorithm by applying optimization concepts of VLSI signal processing such as retiming and parallelism. This facilitates the reduction of critical path and computational time of the algorithm. The algorithm is implemented in Simulink-MATLAB 2013, and the corresponding Verilog codes are written and synthesized in Xilinx ISE Design Suite 14.7. Smith–Waterman algorithm is implemented with reduced critical path and increased speed by applying the VLSI signal processing techniques of pipelining, retiming, and parallelism. The 2-parallel implementation has a critical path of 5Tm +2T and pipelining is performed to reduce the critical path to 3Tm.

Mensah, Bankas, & Iddrisu, (2018) proposed the use of the parallelism and carry-free propagation properties of Residue Number Systems (RNS) to accelerate the algorithm on an FPGA. Linear Systolic Arrays is employed to avoid the use of Lookup Tables with a larger dynamic range and aligns long sequences. The scheme is implemented on a Kintex7 FPGA We propose

The use of the moduli set chosen is used to perform the additions in the SW processing element (PE) on an FPGA to speed-up the computations involved in filling the cells in the alignment matrix of the SWA. It has a dynamic range of 3n bits and preferred over the popular moduli set because the states in the later moduli set requires n+1 bits to represent, leaving almost half of these states unused. Experimental result shows that an approximate of 15 GCUPS and can align long sequences with little degradation in speed. The implementation has a performance improvement of 169 times over the general-purpose processor implementation. This result was obtained using the naïve implementation of LSAs, indicating the positive effects of using RNS.

3.6 Cloud Storage

RRNS has been used to improve cloud storage systems' reliability, confidentiality and security specifications**.** In (Tchernykh et al., 2018)the authors have shown that the known homomorphic cloud computing encryption scheme (HORNS) is not machine safe and efficient and have proposed a Anti-collision method with RRNS.

Gomathisankaran et al., (2011) proposed a homomorphic encryption scheme using Residue Number System (RNS) for cloud storage. In this scheme, a secret is split into multiple shares on which computations can be performed independently. Security is enhanced by not allowing the independent clouds to collude. Efficiency is achieved through the use of smaller shares. In order to prevent such a possibility we want to design the HORNS in such a way that the cloud should be able to operate on the data without having to know the actual modulus One way to add confusion is to transform the modulus by multiplying it with a random noise. One way to protect the confidentiality of modulus is to distribute the computations to different clouds A simple way to distribute the moduli to the clouds is to create disjoint subsets of P and distribute it to all the clouds.

Kar, Sur, Basak, Sukla, & Das, (2016) proposed an effective approach to ensure the data security in cloud computing by the means of changing the binary number system into residue number system (RNS).The most significant part is that we are not using any encryption technique for data hiding. Read the raw input data file (text type) from the user which will be stored in the cloud, the file name of the encrypted data and a coma separated private key (three prime numbers). After reading the input file, we need to prepare the RNS table using the Prepare Table function block that creates the residue moduli set & calculates the summation of each set. All the characters present in the file are converted to their ASCII code so that each Character or each Code is replaced by their respective Residue value present in the Mapping Table. To decrypt Replace each of the Residue number set present in the encrypted file to its corresponding ASCII code in the decrypted file, using the private key set from the prepared RNS table by using the mapping technique.

N. Chervyakov, Babenko, Tchernykh, & Kucherov, (2017) proposed a configurable, reliable, and confidential distributed data storage scheme with the ability to process encrypted data and control results of computations. Redundant Residue Number System (RRNS) with new method of error correction codes and secret sharing schemes. The authors introduced the concept of an approximate value of a rank number (AR), which reduces the computational complexity of the decoding from RNS to binary representation, and size of the coefficients. Based on the properties of the approximate value and arithmetic properties of RNS. The theoretical basis to configure probability of information loss, data redundancy, and speed of encoding and decoding to cope with different objective preferences, workloads, and storage properties was performed by the authors. Theoretical analysis shows that by appropriate selection of RRNS parameters, the scheme allows not only increasing safety, reliability, and reducing an overhead of data storage, but also processing of encrypted data.

Tchernykh et al., (2018) proposed a method to solve the problem of cloud collusion, Based on modified threshold Asmuth-Bloom and Mignotte secret sharing schemes. The authors prove that the algorithm satisfies the formal definition of computational security. When the adversary coalition knows the secret shares, but does not know the secret key, the probability to obtain the secret is very difficult. In AC-RRNS scheme, one RRNS share is stored in one cloud provider. To prevent cloud collusion, a secret key is used. RNS moduli of the size in bits equals to the size of the machine word is used. The properties of RNS error correction codes allow detecting errors in the scheme. The authors demonstrated that the scheme ensures security under several types of attacks.

3.7 Blockchain and IoT

Mei, Gao, Guo, Zhao, & Yang, (2019) proposed a storage mechanism for blockchain architecture where the storage volume on each node is greatly compressed by only storing the remainder of each account to a modulus, and the updating of the account information on each node is performed independently Most of the current solutions would modify the architecture of blockchain, which weakens the characteristics of the decentralization, such as cloud storage. CRT-II (The New Chinese Remainder Theorem) recovery procedure is used to detect damaged data in devil nodes that allow for a strong defect tolerance mechanism in the proposed storage mechanism. Both theoretical analysis and simulation results prove the effectiveness and reliability of the proposed scheme however there is network and overhead computation.

Pandey, Mitharwal, & Karmakar, (2019) proposed a hardware-software co-design implementation of the ECC cipher to provide an adequate level of security for IoTs.. The algorithm is modelled in C language. Computational intensive components are identified for their efficient hardware implementations. Residue number system (RNS) with projective coordinates are utilized for performing the required arithmetic operations. Xilinx platform studio tool and Virtex-5 xc5vfx70t device based platform are used to manage the hardware- software co-design. An application of the implementation is demonstrated for encryption of text and its respective decryption over prime fields. The time-critical functions have been implemented as custom-made components in hardware, the arithmetic operations are converted into a set of coordinates which have been sent to the receiver through the channel in the form of characters so that they cannot be recognized by the eavesdropper. An application has been provided for encryption and decryption that can be useful security in IoTs.

4. CHALLENGES OF RESIDUE NUMBER SYSTEM BASED APPLICATIONS AND RESEARCH DIRECTIONS

It is clear from our literature review that the technology revolution of hardware implementation platforms and design automation tools can be a game changer on the competitiveness of RNS. What used to be the most critical figures of merits for a winning design in the previous decade may no longer be valid today due to the revaluation of cost factors in memory, wiring, power dissipation, thermal, yield and reliability as technologies advance.

- **Selection of Moduli Set**
  From literature, the determination of the best moduli set for various types of application and design is one of the most relevant questions in the RNS. Over the years, a significant number of modules have been introduced, but most are focused on reverse converters. In other RNS based applications such as routing in Software Defined Network, the choice of moduli set determines the dynamic range as the network scales up where the value might not correspond to physically possible paths. Furthermore more, from literature, the moduli set chosen determines both the complexity of arithmetic algorithms and efficiency. There has not been a moduli set with a holistic approach for implementations in an RNS based solution. Major moduli set chosen do have an adverse effect on other sections of the system's output and result in unusable set of modules. Several moduli set have been proposed from literatures in tackling the implementations issues for the well-known three-moduli set $\{ 2^{n-1} - 1, 2^n - 1, 2^n \}$. However, nowadays, due to applications of RNS in artificial intelligence, bioinformatics, network communications, cloud storage there is need for more moduli sets with higher parallelism and dynamic range to achieve optimum performance.

- **Overhead problems due to Conversions**
  Overhead issues emerging from reverse and forward transformation calculations and how they can be used for various applications, though successful hardware implementation remains a challenge. Various RNS components, such as modular arithmetic's circuits, transformations, overflow and sign-detectors, residue and scaling units, increase the complexity of complete RNS which makes the application difficult to use. Here, one solution is using hybrid architectures. For example, a common adder with a dual output can be designed instead of designing separate adders for modules 2^{n-1} and $2^n - 1$. This approach can also be used to combine other system components. The hybrid hardware can also be used to implement various components.

- **Hardware Implementations**
  The RNS's hardware architecture is an important factor that affects the wide adoption and implementation of RNS. Additionally, the hardware implementation of an RNS based application is also influenced by the moduli set chosen. From literature, In general, RNS is not designed for programmable processor implementation due to its transcoding overhead, therefore, more research is needed in this area.

CONCLUSIONS AND FUTURE DIRECTIONS

In order to assist potential researchers to have a preliminary knowledge of the area, we presented a detailed RNS field study and applications. Moduli selection and data transformations were well discussed since they are the major bottleneck in the adoption of RNS. We also presented the areas of applications of RNS in various computer related fields. As challenges for the future, we will investigate the applicability of the field (RNS) to improving other areas of RNS based applications.
Today, however, we need more parallel and dynamic moduli sets to achieve critical efficiency, so as to meet requirements of secure, stable and scalable systems. We have designed almost all RNS scaling, sign detection and residue comparison circuits for the well-known 3 module set $\{ 2^{n-1} - 1, 2^n - 1, 2^n \}$. The application on hardware and powerful moduli set to exploring the multi-core CPU and GPU is an important research direction for the future.
RNS can still further be applied to other applications that require complex calculations such as Algorithms for Network Intrusion detection Systems, Security, performance and scalability of Blockchain systems, Cryptographic mixing algorithms to ensure Privacy in Voting and Contact Tracing app (like Covid-19 Tracing applications) etc. There still exist several areas to be explored in the use of RNS for building simpler core network elements for package forwarding in Software Defined Networking and Wireless Sensor Networks. The stateful requirement for the active flows that imposes scalability, responsiveness, cost and power consumption problems for the application of SDN in core networks can be tackled with RNS computation.

REFERENCES

[1] **Abdelfattah, O.** (2011). *Data Conversion in Residue Number System*. 114.

[2] **Abdelhamid, M., & Koppula, S.** (2017). *Applying the Residue Number System to Network Inference*.

[3] **Agbedemnab, P. A., Yellakuor, E., & Daabo, M. I.** (2019). *Single and Multiple Error Detection and Correction using Redundant Residue Number System for Cryptographic and Stenographic Schemes*. *4*(4), 1–14. https://doi.org/10.9734/AJRCOS/2019/v4i430123

[4] **Ahmadi, K., & Salari, E.** (2014). *An Image Hiding Algorithm Using Chinese Remainder Theorem with Reduced Distortion*. 240–245.

[5] **Alhassan, I. Z., & Ansong, E. D.** (2020). *Enhancing Image Security during Transmission using Residue Number System and k-shuffle*. *4*(2), 399–424.

[6] **Alhassan, S., & Gbolagade, K. A.** (2013). *Enhancement of the Security of a Digital Image using the Moduli Set*. *2*(7), 2223–2229.

[7] **Amine, B. M., & Mamoun, S. El.** (2019). *Introduction to Steganography in RRNS based Communications*.

[8] **Ammar, A., Ai, A. W. B. A. N. Y., Youssef, M., & Emam, A.** (2001). *A Secure Image Coding Scheme using Residue Number System*. 399–405.

[9] **Aremu, I. A., & Gbolagade, K. A.** (2017). *An overview of Residue Number System*. *6*(10), 1618–1623.

[10] **Atta-Ur-Rahman, Naseem, M. T., Qureshi, I. M., & Muzaffar, M. Z.** (2011). Reversible watermarking using Residue Number System. *Proceedings of the 2011 7th International Conference on Information Assurance and Security, IAS 2011*, (December 2015), 162–166. https://doi.org/10.1109/ISIAS.2011.6122813

[11] **Azizifard, A., Qermezkon, M., & Farshidi, R.** (2015). *Information Steganography within 3D Images Using Residue Number System*. (February 2015).

[12] **Azizifard, A., Qermezkon, M., Postizadeh, T., & Barati, H.** (2014). *Data Steganography on VoIP through Combination of Residue Number System and DNA Sequences*. *5*(2), 7–22.

[13] **Baagyere, E. Y., Agbedemnab, P. A. N., Qin, Z., Daabo, M. I., & Qin, Z.** (2020). A Multi-Layered Data Encryption and Decryption Scheme Based on Genetic Algorithm and Residual Numbers. *IEEE Access*, *8*, 100438–100447. https://doi.org/10.1109/ACCESS.2020.2997838

[14] **Baagyere, Y. E.** (2011). *Application of residue number system to smith-waterman algorithm*.

[15] **Babatunde, Akinbowale N, Jimoh, R. G., & Gbolagade, K. A.** (2016). An algorithm for a residue number system based video encryption system. *Annals. Computer Science Series. 14th Tome 2nd Fasc. – 2016*, *XIV*, 137–145.

[16] **Babatunde, Akinbowale Nathaniel.** (2019). Methodology for Image Cryptosystem Based on a Gray Code Number System. *School of Computing, Engineering & Physical Sciences Computing and Information Systems Journal Vol 23, No 2, 2019*, *23*(2).

[17] **Babatunde, Akinbowale Nathaniel, Jimoh, E. R., Oshodi, O., & Alabi, O. A.** (2019). Performance analysis of gray code number system in image security. *Jurnal Teknologi Dan Sistem Komputer*, *7*(4), 141–146. https://doi.org/10.14710/jtsiskom.7.4.2019.141-146

[18] **Babenko, M., E, S., Tchernykh, A., & Golimblevskaia.** (2020). S ingle b ase e xtension (SBE). *The Dictionary of Genomics, Transcriptomics and Proteomics*, 1–1. https://doi.org/10.1002/9783527678679.dg11943

[19] **Bajard, J. C., Duquesne, S., Ercegovac, M., & Meloni, N.** (2006). *Residue systems efficiency for modular products summation : Application to Elliptic Curves Cryptography*. *6313*, 1–11. https://doi.org/10.1117/12.679541

[20] **Belhamra, M. A., & Souidi, E. M.** (2020). *Journal of Information Security and Applications Steganography over Redundant Residue Number System Codes*. *51*. https://doi.org/10.1016/j.jisa.2019.102434

[21] **Bhangale, P. P., Raje, R. S., Maurya, J., & Gawad, A.** (2017). *Image Security using AES and RNS with Reversible Watermarking*. *4*(5), 345–349.

[22] **Campobello, G., Leonardi, A., Palazzo, S., & Member, S.** (2012). *Improving Energy Saving and Reliability in Wireless Sensor Networks Using a Simple CRT-Based Packet-Forwarding Solution*. *20*(1), 191–205.

[23] **Cardarilli G C, Re A., Del Lojacono R., Nannarelli A., Re M., Politecnico, V.** (2003). *RNS Implementation of High Performance Filters for Satellite Demultiplexing*.

[24] **Cardarilli Gian Carlo, Nannarelli, A., & Re, M.** (2007). *Residue Number System for Low-Power DSP Applications*. (1), 1412–1416.

[25] **Chang, C.** (2015). *Residue Number Systems : A New Paradigm to Datapath Optimization for Low-Power and Digital Signal Processing Applications*.

[26] **Chervyakov, N., Babenko, M., Tchernykh, A., & Kucherov, N.** (2017). AR-RRNS : Configurable reliable distributed data storage systems for Internet of Things to ensure security. *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2017.09.061

[27] **Chervyakov, N I, Lyakhov, P. A., Deryabin, M. A., Nagornov, N. N., Valueva, M. V, & Valuev, G. V.** (2020). Residue Number System-

Based Solution for Reducing the Hardware Cost of a Convolutional Neural Network. *Neurocomputing*. https://doi.org/10.1016/j.neucom.2020.04.018

[28] **Chervyakov, Nikolai I, Lyakhov, P. A., Nikolai, N., & Bogayevskiy, V.** (2019). *Implementation of Smoothing Image Filtering in the Residue Number System*. (June), 8–11.

[29] **Del Re, A., Nannarelli, A., & Re, M.** (2002). *Implementation of digital filters in carry-save residue number system*. 1309–1313 vol.2. https://doi.org/10.1109/acssc.2001.987702

[30] **Deryabin, M., Chervyakov, N., & Tchernykh, A.** (2018). *High Performance Parallel Computing in Residue Number System High Performance Parallel Computing in Residue Number System*. *9*(February), 62–67.

[31] **Eseyin, J. B.** (2019). *An Overview of Public Key Cryptosysems and Application of Residue Number System*. *4*(2), 37–44.

[32] **Eseyin, J. B., & Gbolagade, K. A.** (2019). *A Residue Number System Based Data Hiding Using Steganography and Cryptography*. *5*(2), 345–351.

[33] **Fournaris, A. P., & Sklavos, N.** (2016). *Residue Number System as a Side Channel and Fault Injection Attack countermeasure in Elliptic Curve Cryptography*.

[34] **Fred, T. J.** (1990). An RNS Discrete Fourier Transform Implementation. *IEEE transactions on acoustics, speech. And signal processing, VOL. 38. NO. 8. AUGUST 1990*, *38*(8), 1386–1394.

[35] **Gbolagade, K. A.** (2010). *Effective Reverse Conversion in Residue Number System Processors*.

[36] **Gomathisankaran, M., Tyagi, A., & Namuduri, K.** (2011). HORNS: A homomorphic encryption scheme for cloud computing using residue number system. *2011 45th Annual Conference on Information Sciences and Systems, CISS 2011*. https://doi.org/10.1109/CISS.2011.5766176

[37] **Gomes, R. R., Liberato, A. B., Dominicini, C. K., Ribeiro, M. R. N., & Martinello, M.** (2016). KAR: Key-for-Any-Route, a Resilient Routing System. *Proceedings - 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN-W 2016*, 120–127. https://doi.org/10.1109/DSN-W.2016.11

[38] **Hiasat, A.** (2017). An Efficient Reverse Converter for the Three-Moduli Set (2n+1-1, 2n, 2n-1). *IEEE Transactions on Circuits and Systems II: Express Briefs*, *64*(8), 962–966. https://doi.org/10.1109/TCSII.2016.2608335

[39] **Hiasat, A.** (2019). A Reverse Converter for Three-Moduli Set. *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 548–553.

[40] **Junior, J. A., Fernando, L., Nascimento, L., Carlos, L., & Albini, P.** (2011). *Using the Redundant Residue Number System to increase Routing Dependability on Mobile Ad Hoc Networks*. 67–73.

[41] **Kar, A., Sur, K., Basak, S., Sukla, A. S., & Das, R.** (2016). *Securiity in cloud storage : An Enhanced Technique of Data Storage in Cloud using RNS*. 6–9.

[42] **Kayode, S., & Gbolagade, K. A.** (2017). *Efficient RSA cryptosystem decryption based on chinese remainder theorem and strong prime*. *XV*, 1–5.

[43] **Kehinde, H., & Alagbe, K.** (2018). Residue Number System: An Important Application in Bioinformatics. *International Journal of Computer Applications*, *179*(10), 28–33. https://doi.org/10.5120/ijca2018916106

[44] **Kenneth, J. W., & J., B. L.** (1977). *The Use of Residue Number Systems Response Digital Filters*.

[45] **Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S.** (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, *103*(1), 14–76. https://doi.org/10.1109/JPROC.2014.2371999

[46] **Lab, R.** (1994). *High-Speed RSA Implementation*. (November).

[47] **Liberato, A., Martinello, M., Gomes, R. L., Beldachi, A. F., Salas, E., Villaca, R., Simeonidou, D.** (2018). *Residue Defined Networking Architecture*. *15*(4), 1473–1487.

[48] **Liberato, A., Martinello, M., Gomes, R. L., Beldachi, A. F., Salas, E., Villaca, R., … Simeonidou, D.** (2018). RDNA: Residue-Defined Networking Architecture Enabling Ultra-Reliable Low-Latency Datacenters. *IEEE Transactions on Network and Service Management*, *15*(4), 1473–1487. https://doi.org/10.1109/TNSM.2018.2876845

[49] **Lim, Z., & Phillips, B. J.** (2007). An RNS-enhanced microprocessor implementation of public key cryptography. *Conference Record - Asilomar Conference on Signals, Systems and Computers*, 1430–1434. https://doi.org/10.1109/ACSSC.2007.4487465

[50] **Luan, Z., Chen, X., Ge, N., & Wang, Z.** (2014). *Simplified fault-tolerant FIR filter architecture based on redundant residue number system*. *50*(23), 1768–1770. https://doi.org/10.1049/el.2014.3508

[51] **Madhukumar, A. S., Chin, F., & Premkumar, A. B.** (2000). Residue number system based multicarrier CDMA for broadband mobile communication systems. *Midwest Symposium on Circuits and Systems*, *2*, 536–539. https://doi.org/10.1109/MWSCAS.2000.952812

[52] **Martinelli, G., Perfetti, R., Universitk, I. D., & Sapienza, L.** (1990). *RNS NEURAL NETWORKS G. Martinelli, R. Perfetti INFO-COM Dpt. Universitk di Roma "La Sapienza" via Eudossiana, 18 - 00184 Roma - Italy*. 2955–2958.

[53] **Martinello, M., Ribeiro, M. R. N., Oliveira, R. E. Z. De, & Vitoi, D. A.** (2014). *KeyFlow: A Prototype for Evolving SDN Toward Core Network Fabrics*. (April), 12–19.

[54] **Mei, H., Gao, Z., Guo, Z., Zhao, M., & Yang, J.** (2019). *Storage Mechanism Optimization in Blockchain System Based on Residual Number System*. *XX*. https://doi.org/10.1109/ACCESS.2019.2934092

[55] **Mensah, P. K., Bankas, E. K., & Iddrisu, M. M.** (2018). RNS Smith-Waterman Accelerator based on the moduli set 2 n , 2 n-1 , 2 n-1 -1. *2018 IEEE 7th International Conf. on Adaptive Science & Technology (ICAST)*, (1), 1–8. https://doi.org/10.1109/ICASTECH.2018.8506912

[56] **Mohan, P. V. A.** (2002). *Residue Number Systems*.

[57] **Naseem, M., Qureshi, I., Muzaffar, M., & ur Rahman, A.** (2016). Spread spectrum based invertible watermarking for medical images using RNS and Chaos. *International Arab Journal of Information Technology*, *13*(2), 223–231.

[58] **Naseem, M. T., & Muzaffar, M. Z.** (2012). Reversible and Robust Watermarking using Residue Number System and Product Codes. *Journal of Information Assurance and Security*, (December 2015).

[59] **Navi, K., Molahosseini, A. S., & Esmaeildoust, M.** (2011). *How to Teach Residue Number System to Computer Scientists and Engineers*. *54*(1), 156–163.

[60] **Nazarov, A., & Chervyakov, N.** (2018). *Reliability Improvement of Information Systems by Residue Number System Code*. *9*(1), 81–84.

[61] **Omondi, A., & Premkumar, B.** (2007). *Residue Number Residue Number*.

[62] **Oyinloye, D. P., & Gbolagade, K. A.** (2018). An Improved Image Scrambling Algorithm Using {2n -1, 2n, 2n +1}. *School of Computing, Engineering & Physical Sciences Computing and Information Systems Journal Vol 22, No 3, 2018*, *22*(3).

[63] **Pandey, J. G., Mitharwal, C., & Karmakar, A.** (2019). An RNS implementation of the elliptic curve cryptography for IoT security. *Proceedings - 1st IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA 2019*, 66–72. https://doi.org/10.1109/TPS-ISA48467.2019.00017

[64] **Patra, J. C., Kishore, A. K., & Bornand, C.** (2011). *Improved CRT-based DCT Domain Watermarking Technique with Robustness Against JPEG Compression for Digital Media Authentication*. 2940–2945.

[65] **Pettenghi, H., Chaves, R., & Sousa, L.** (2013). RNS reverse converters for moduli sets with dynamic ranges up to (8n+1)-bit. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *60*(6), 1487–1500. https://doi.org/10.1109/TCSI.2012.2220460

[66] **Phalguna, P. S., Kamat, D. V., & Ananda Mohan, P. V.** (2018). RNS-to-Binary Converters for New Three-Moduli Sets {2k - 3, 2k - 2, 2k - 1} and {2k + 1, 2k + 2, 2k + 3}. *Journal of Circuits, Systems and Computers*, *27*(14), 1–20. https://doi.org/10.1142/S0218126618502249

[67] **Phalguna, P. S., Kamat, D. V., & Mohan, P. V. A.** (2019). Novel RNS-to-binary converters for the three-moduli set {2m 2 1, 2m, 2m +1}. *Sadhana - Academy Proceedings in Engineering Sciences*, *44*(4), 1–10. https://doi.org/10.1007/s12046-019-1078-0

[68] **Pontarelli, S., Cardarilli, G. C., Re, M., & Salsano, A.** (2008). *Totally Fault Tolerant RNS based FIR Filters*. 192–194. https://doi.org/10.1109/IOLTS.2008.14

[69] **Priyanka, V., Nireesha, M., Kumar, V. V., Ram, N. V., & Chakravarthy, A. S. N.** (2012). *CRT and ART Based Watermarking Scheme in DCT Domain*. (4), 87–90.

[70] **Qureshi, I. M., & Muzaffar, Z.** (2016). *Spread Spectrum based Invertible Watermarking for Medical Images using RNS & Chaos*. (March).

[71] **Rahman, A. U., Sultan, K., Musleh, D., Aldhafferi, N., Alqahtani, A., & Mahmud, M.** (2018). Robust and Fragile Medical Image Watermarking: A Joint Venture of Coding and Chaos Theories. *Journal of Healthcare Engineering*, *2018*. https://doi.org/10.1155/2018/8137436

Rajalakshmi, K., & Nivedita, R. (2018). VLSI implementation of Smith–Waterman algorithm for biological sequence scanning. *Lecture Notes in Electrical Engineering*, *453*, 231–245. https://doi.org/10.1007/978-981-10-5565-2_21

[72] **Raji, K. A., Gbolagade, K. A., & Taofeek-ibrahim, F. A.** (2018). *An Enhanced Vitality Efficient and Reliable Wireless Sensor Networks with CRT-Based Packet Breaking Scheme*. *6*(2), 26–37. https://doi.org/10.11648/j.ijssn.20180602.11

[73] **Salamat, S., Imani, M., Gupta, S., & Rosing, T.** (2019). RNSnet: In-Memory Neural Network Acceleration Using Residue Number System. *2018 IEEE International Conference on Rebooting Computing, ICRC*, 1–12. https://doi.org/10.1109/ICRC.2018.8638592

[74] **Samimi, N., Kamal, M., Afzalli-kusha, A., & Pedram, M.** (2019). Res-DNN : A Residue Number System-Based DNN Accelerator Unit. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *PP*, 1–14. https://doi.org/10.1109/TCSI.2019.2951083

[75] **Schinianakis, D. M., Kakarountas, A. P., & Stouraitis, T.** (2006). *A New Approach to Elliptic Curve Cryptography: an RNS Architecture*. 1241–1245.

[76] **Schinianakis, D., & Stouraitis, T.** (2016). *Residue Number Systems in Cryptography: Design, Challenges, Robustness*. https://doi.org/10.1007/978-3-319-14971-4

[77] **Shahana, T. K., Jose, B. R., Jacob, K. P., & Sasi, S.** (2008). *RRNS-Convolutional Concatenated Code for OFDM based Wireless Communication with Direct Analog-to-Residue Converter*. *35*(November).

[78] **Shrimali, D., & Sharma, L.** (2018). *An Extensive Review on Residue Number System for Improving Computer Arithmetic Operations*. 1617–1621.

[79] **Singh, N.** (2016a). *An overview of Residue Number System*. (August).

[80] **Singh, N.** (2016b). *An overview of Residue Number System*. (November 2008).

[81] **Singh, P.** (2016). An Efficient CRT based Digital Image Watermarking using Double Density Wavelet Transform. *International Journal of Multimedia and Image Processing*, *6*(1/2), 805–812. https://doi.org/10.20533/ijmip.2042.4647.2016.0041

[82] **Singh, T.** (2014). *Residue Number System for Fault Detection in Communication Networks*. 157–161.

[83] **Soderstrand, M., Jenkins, W., Jullien, G., & Taylor, F.** (1986). Residue number system arithmetic: modern applications in digital signal processing.

[84] **Soudris, DSgouropoulos, K., Tatas, K., & Padidis, V.** (2003). *A Methodology for Implementing FIR Filters and CAD Tool Development for Designing RNS-Based Systems*. (1), 129–132.

[85] **Sousa, L.** (2007). Efficient method for magnitude comparison in RNS based on two pairs of conjugate moduli. *Proceedings - Symposium on Computer Arithmetic*, 240–247. https://doi.org/10.1109/ARITH.2007.16

[86] **Taleshmekaeil, D. K., & Mousavi, A.** (2010). The use of Residue Number System for improving the Digital Image Processing. *International Conference on Signal Processing Proceedings, ICSP*, 775–780. https://doi.org/10.1109/ICOSP.2010.5655920

[87] **Tchernykh, A., Babenko, M., Chervyakov, N., & Miranda-lópez, V.** (2018). *AC-RRNS : Anti-Collusion Secured Data Sharing Scheme for Cloud Storage*. 137–141. https://doi.org/10.1109/DEXA.2017.44

[88] **Thabah, S. D., Sonowal, M., & Saha, P.** (2018). On the Design of Efficient Residue-to-Binary Converters. *Procedia Computer Science*, *132*, 816–823. https://doi.org/10.1016/j.procs.2018.05.093

[89] **Toivonen, T.** (2006). *Video Filtering With Fermat Number Theoretic Transforms Using Residue Number System*. *16*(1), 92–101.

[90] **Valentim, R. V., Villaca, R. S., Ribeiro, M. R. N., Martinello, M., Dominicini, C. K., & Mafioletti, D. R.** (2019). *RDNA Balance: Load Balancing by Isolation of Elephant Flows using Strict Source Routing*. 1–3.

[91] **Valueva, M., Valuev, G., Semyonova, N., Lyakhov, P., Chervyakov, N., Kaplun, D., & Bogaevskiy, D.** (2019). *Construction of Residue Number System Using Hardware E ffi cient Diagonal Function*. 1–16.

[92] **Vassalos, E., & Bakalis, D.** (2013). *RNS Assisted Image Filtering and Edge Detection*.

[93] **Wei Wang, Swamy, M. N. S., & Ahmad, M. O.** (2004). *RNS application for digital image processing*. (August), 77–80. https://doi.org/10.1109/iwsoc.2004.1319854

[94] **Yatskiv, V., Sachenko, A., Nataliya, Y., Bykovvy, P., & Segin, A.** (2019). Compression and Transfer of Images in Wireless Sensor Networks Using the Transformation of Residue Number System. *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, *2*, 1111–1114.

[95] **Yatskiv, V., & Tsavolyk, T.** (2017). *Improvement of Data Transmission Reliability in Wireless Sensor Networks on The Basis of Residue Number System Correcting Codes Using the Special Module System*. *1*, 890–893.

[96] **Zhi-Gang, L., & Mattina, M.** (2020). *Efficient Residue Number System Based Winograd Convolution*.