Semiconductor Optoelectronics
ISSN:
1001-5868

EXPLORING THE RESIDUE NUMBER SYSTEM VIA CONJUGATE MODULI SETS

Bentipalli Sekhar[1], G Appala Naidu[2], K. Babulu[3],
[1]Research scholar, JNTU-Gurajada-Vizianagaram College of Engineering, India
[2]Assistant Professor in ECE, JNTU-Gurajada-Vizianagaram College of Engineering, India
[3]Professo in ECE, JNTU-Gurajada-Vizianagaram College of Engineering, India

Abstract□— In recent years, electronics & communication and computer applications like, Data Communication and Networking, Digital Signal Processing, FIR filters, Artificial Intelligence, Bioinformatics, and Cryptography…etc. be in need of fast computation with error free operations required. To furnish these, different parameters matter, in the midst of the number system used will also play a major role. Residue Number System (RNS) [1], which has carry free in nature and fault tolerant discharges the above-mentioned operations with go over. The significant issues, in effective plan of RNS based system are the moduli set selection, forward conversion i.e. Analog/Digital number to RNS representation, residue arithmetic unit and reverse conversion i.e. RNS representation to Analog/Digital number

In this paper, conjugate moduli sets are selected for analyzing how selection of moduli sets affects the performance of RNS. Considered moduli sets are [2], special moduli set {2n-1, 2n, 2n+1}, {2n-1, 2n, 2n+1, 2n+1-1} and [3], conjugate moduli set {2 n1-1, 2 n1+1, 2 n2-1, 2 n2+1}. .For the above mentioned, date conversion examples and also one of the applications [4], of RNS i.e. FIR fiters with RNS also investigated. From [5, 3], performances of selected moduli sets are reviewed, compared and summarized. It is concluded that, conjugate moduli set {2 n1-1, 2 n1+1, 2 n2-1, 2 n2+1} has better performance among the selected conjugate moduli sets and its utility in the above-mentioned applications yields greater impact on performance of the RNS based system.

Index Terms— Residue number system, conjugate moduli sets, data conversion, and RNS arithmetic.

INTRODUCTION

Speed and power consumption are major main features for current and future systems, particularly portable platforms [6],[7]. Parallelism may be used at various degrees of electronic system design in this criterion, including a variety of architectures & algorithms to execute arithmetic calculations concurrently on various more modest parts, allowing designers to reduce power consumption and/or improve performance [8]. While much progress has been achieved in terms of design and implementation, the most of devices often struggle with the drawbacks associated with weighted binary number systems. These number schemes involve a long carry-propagation chain, which is inefficient for dealing with the produced power and latency. This problem prompted the study of different numbering schemes such as residue number systems (RNS) [9,10]. The residue number scheme is a non-weighted number system. As such, it is noticeable from the binary or decimal number systems, as well as the weighted number system. Inherently residue arithmetic operations i.e. basic mathematical operations except division operation, in which each digit of the output is a function of a single variable in each operand, unfettered of the other digits. This function often results in a significant boost in processing capacity, which is the primary concern in DSP applications.

When addition and multiplication are the primary arithmetic operations, RNS is suitable. RNS holds great promise in situations where speed and/or power consumption are crucial thanks to its carry-free characteristic. Separating modulo channels also makes it possible to spot errors and correct them. These technologies comprise contact receivers [14], fault tolerance [15], RSA algorithms [13], digital signal processing (DSP) [11], and digital image processing (DIP) [12]. These applications generally require extensive multiply-and-accumulate (MAC) procedures.

The creation of digital filters is one potential application of RNS in DSP. Interpolation, decimation, equalisation, noise reduction, and band separation are just a few of the many uses for digital filters. Finite impulse response (FIR) filters and infinite impulse response (IIR) filters are the two different categories of digital filters (IIR). The system is expedited and its power consumption is decreased by performing the necessary multiplication and addition operations in the residue domain [16,17]. The Discrete Fourier Transform (DFT), which is widely used in many engineering applications, is another potential application of RNS in DSP. Again, the crucial operations in this situation are addition and multiplication. RNS is used to run DFT algorithms, and this produces results because of the parallelism built into processing. The RNS has no information on weight. The other modulo channels are likewise unaffected by any error in a residue. Additionally, since ordering is irrelevant in RNS representation, it is feasible to reject and repair faulty residues individually. In summary, RNS seems be sufficient for an enormous number of utilizations essential to contemporary computing methods.

General Architecture of RNS Processor

The basic design of a typical RNS processor is depicted in Figure 1. The data conveyed by the RNS is analyzed in a linear pattern, with no dependencies or propagation between processing units.

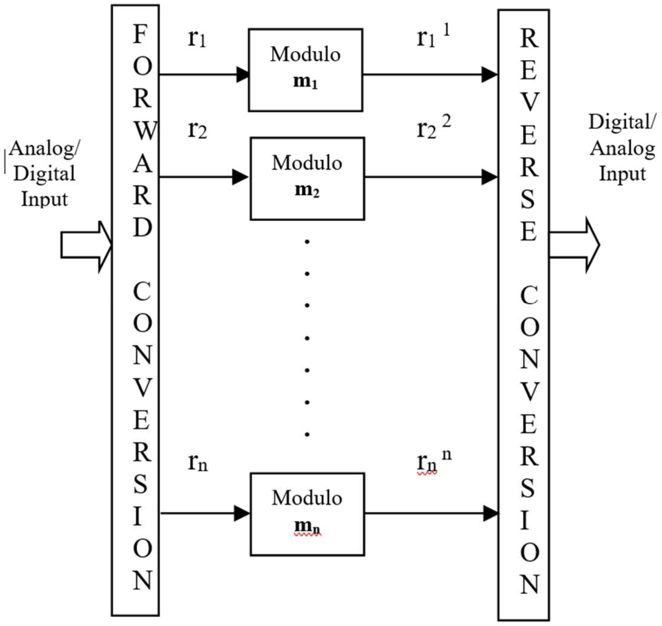

Fig. 1. Generalized architecture of RNS based system.

Forward Conversion is the process of changing input data into RNS representation, while Reverse Conversion is the method of changing RNS output data back to regular format. The conversion phases are critical in determining the general RNS's efficiency. Conversion circuitry may be very complex, and delay can be introduced that compensates for the RNS processors' speed. The connection with the analogue environment encompasses the conversion of analogue to residue and vice versa for a fully RNS reliant device. This is usually accomplished in two steps, with the binary conversion serving as an intermediate step. As a result of their increasing latency and complexity, the conversion step becomes inefficient. We need to develop conversion circuits that perform as well as digital signal processing applications in digital binary-based systems in order to develop an RNS processor that may be used in lieu of a digital processor in certain applications. The reverse conversion method depends on the Chinese Remainder Theorem (CRT) or Mixed-Radix Conversion (MRC) approaches. Investigating novel conversion methods will enable the RNS to overcome these roadblocks in the implementation of different applications. To eliminate the intermediary binary stage [18], we are pursuing an analogue-to-residue (A/R) converter and a residue-to-analog (R/A) converter.

LITERATURE REVIEW

Jaberipur Ghassem et al[19] addressed that the forward translation of the binary operands, the resulting residues of the RNS and significant cost was imposed by translating the RNS's effects back to its binary operands. RNS arithmetic won't be helpful, then, unless several residue operations happen in succession before the requirement to convert a result back to binary arises. Overall, it is typically preferable to attain, roughly, the same speed for all residues through modulus determination for the same procedures.

$$m_i = 2^n - \delta,$$

Where $0 \leq \delta \leq 2^{n-1}$.

Azadeh Alsadat Emrani Zarandi et al[20] demonstrated another high level viper part to deliver turn around converter structures capable of supplying negative RNS numbers with two complementary representations. This adder part is modular and can easily be added to any c-class module-set reverse converter by simply replacing the final module adder. The development of application-specific integrated circuits is greatly facilitated by the suggested component, according to theoretical evaluations and assessments, which only slightly increase the overhead of standard unsigned reverse converters. The suggested architecture improves on the state of the art for producing signed output converters by up to 9 percent, 21 percent, and 35 percent, respectively, in terms of delay, chip area, and energy usage. Through the creation of efficient reverse conversion structures, this work constitutes a significant milestone in the implementation of signed RNSs.

Zeinab Torabi et al[21] have shown that RNS number comparisons are largely more expensive and time-consuming than standard RNS arithmetic operations, such as adding and multiplying. That is why comparison-free computations are the most popular RNS applications. Some exploration endeavors, however, turned out to be flexible to achieve successful RNS comparators, primarily for the common moduli set τ = {2n-1, 2n,2n +1}. Our contribution is the latest τ -comparator that uses a novel approach based on the dynamic range partitioning and an integer interval mapping of the partitions. Thus, the operation of comparison is limited to matching the partition numbers corresponding to the operands of comparison. This enables the two adders to be merged into one parallel prefix compound adder, which dramatically decreases the cost and dissipation of electricity. The same choice of compounding does not extend to the previous realizations of τ -comparator that still depend on two modules (2n−1) adders Post layout simulation findings confirm our thorough theoretical assessment of the proposed design and the best prior one, which greatly favours ours in terms of area use. Improvements can be summarized as a decrease in area usage of 17 percent (46 percent), power dissipation of 30 percent (41 percent) and PDP of 31 percent (47 percent), with a slight latency benefit of 0.7 percent (10 percent), for n = 8 (22). We plan to extend our current methodology to widely used four- and five-moduli RNS sets and in the design of RNS dividers for future applicable studies.

Piotr Patronik and Stanisaw J. Piestrak et al. proposed a novel architecture strategy for an enhanced processor arithmetic unit, introduced utilizing the residue number system (RNS)[22]. In addition, unlike any previous design, we suggested a hardware/software hybrid solution, in which a few fundamental modular operations are implemented in hardware and reverse conversion is implemented in software, allowing for a reduction in static power consumption due to the latter circuit. Some of the measurement channels are completely disabled as a result of the program's reverse conversion execution and the underlying RNS datapath partitioning. The latter not only reduces power consumption and execution time in standard RNS implementations, but also makes the solution shown here interesting for implementations involving complex increases in the precision of the calculations. In comparison to executions using a positional arithmetic unit, the results of the power simulation demonstrate that RNS arithmetic units have smaller area and delay due to smaller modular multipliers, allowing up to more than 20% energy savings for each of the three tested applications: constant coefficient filtering, matrix multiplication, and large Montgomery multiplication. Additionally, the suggested method is simple to integrate into software compilers, enabling a programmer able to see (at least a portion of) RNS implementation. The study of several aspects of the intricate shift in numerical accuracy of the proposed solution will also be included in further analysis.

RNS representation

An RNS is determined by the moduli, a collection of comparatively prime integers. The moduli-set is abbreviated as {m1,m2,m3,……mn} with "mi" denoting the ith modulus. Each integer can be thought of as a collection of smaller integers, referred to as residues. The label for the residue set reads {r1,r2,r3,……rn},where "ri" stands for the ith modulus. The residue ri is known as the least positive residue when divided by the modulus mi.

When divided by the modulus mi, the residue ri is defined as the least positive remainder. On the basis of congruence, this relationship can be written notationally:

$X \, mod \, mi = ri$ …. ………………………… (1)

In an alternate notation, the same coherence can be written as:

$|X|_{mi} = r_i$ ……………………………….… (2)

Both integers X which reside in its dynamic spectrum can be uniquely constituted by the RNS. The moduli-set {m1,m2,m3,……mn} defines the dynamic spectrum and denotes it as M

Where,

$M = \prod_{i=1}^{n} mi$ ....……...…………….................. (3)

The RNS provides specific representations for all numbers in the range 0 to M-1. If the integer X is bigger than M-1, the RNS representation repeats. The representation of residues might then be the same for multiple integers.

It is crucial to emphasize that the moduli must be comparatively prime in order to be able to use the maximum dynamic range M.

We believed that unsigned integers were the main topic of concern in the prior RNS discussion. Negative numbers are however represented in some applications. In order to do this, we should divide the entire range [0: M-1] into two nearly equal halves: the top half represents the positive numbers, while the lower half represents the negative integers. The relationships [23] below must be met by the numbers X that can be described by the existing convention:

$$-\frac{M-1}{2} \leq X \leq \frac{M-1}{2} \; if \; M \; is \; odd$$

$$-\frac{M}{2} \leq X \leq \frac{M}{2} - 1 \; if \; M \; is \; even$$

If X={r_1,r_2,r_3,……r_n} represents a positive number inside the proper range, then -X can be written as {r_1',r_2',r_3',……r_n'} where ri' is the complement of r_i for the given value of m_i, where r_i' satisfies the relation (r_i+r_i') mod m_i=0.

1) Mathematical representation

The foundations of RNS representation are covered in this section. The comparisons are described in-depth utilizing their traits. These qualities offer a solid framework for thinking about the conversion process between the traditional approach and the RNS. You will find more complex findings and statistical linkages on the following pages. This section talks about RNS's fundamental algebra. Finding the inverses of the additive and multiplicative functions as well as various division and scaling properties that are not covered by the basic RNS operations are required for this.

Residue of a number:

The essential relationship between numbers in conventional representation and RNS representation is the following congruence:

$X\ mod\ mi = ri$……………..… (4)

Where, m_i is the modulus and r_i is the residue. The residue is known as the least positive remainder when the integer X is divided by the modulus m_i.

Definition of the base values:

Whatsoever number X can be represented as a combination of the base value B_i and residue r_i for the modulus m_i.

X= B_i+r_i ………………………………. *(5)*

B_i=k x $m_{..i}$ ……………………………….. *(6)*

Where X is an integer which is in Equation

Addition (or subtraction)

We can add (or subtract) different integers into the RNS representation by separately adding (or removing) the residues with respect to the appropriate moduli.

Consider the S={m_1,m_2,m_3, ... m_n} module group. The RNS representation includes the X and Y numbers.

Consider, X={x_1,x_2,……..x_n} & Y={y_1,y_2,……..y_n}

Now, Z=X+Y= {z_1,z_2,……..z_n}

Corresponding z_i calculated as,

z_i=(x_i+y_i) mod m_i ……………………….(7)

This characteristic can be used to subtraction as well, where the addition of Y is taken into account when Y is subtracted from X.

Modulo is a distributive operation compared to addition and subtraction:

$\left| X \pm Y \right|_m = \left| \left| X \right|_m \pm \left| Y \right|_m \right|_m$ …..................... (8)

Multiplication

By multiplying the various residues with respect to their respective moduli, RNS multiplication can be accomplished in a manner similar to addition.

The X and Y integers in the RNS representation are given, and we should consider the S={m_1,m_2,m_3, ... m_n} moduli set.

Consider, X={x_1,x_2,……..x_n} & Y={y_1,y_2,……..y_n}

Now, Z=X*Y= {z_1,z_2,……..z_n}

Corresponding z_i calculated as,

z_i=(x_i*y_i) mod m_i ……………………….…… (9)

Over multiplication, the modulo operation is distributive.

$\left| \mathbf{X * Y} \right|_{\mathbf{m}} = \left| \left| \mathbf{X} \right|_{\mathbf{m}} * \left| \mathbf{Y} \right|_{\mathbf{m}} \right|_{\mathbf{m}}$ …..........……...... (10)

Division

The separation is one of the main obstacles to reducing the use of RNS. Division is a difficult operation in RNS representation. Traditional representation division and RNS representation cannot be compared.

We portray division in traditional representation as follows:

$\boldsymbol{\frac{X}{Y} = q}$ .....…………………. (11)

That can be rephrased as: y X q=x, here q is quotient

In RNS, the analogous congruence is:

Y X q = x mod m ……………………...….. (11')

By multiplying both sides by the multiplicative inverse of y, we can write:

$\boldsymbol{q = X * Y^{-1}\ mod\ m}$ ……… (12)

Additive Inverse

The relationship between the residue r_i and its additive inverse r_i' is determined by congruence.

$(r_i+r_i^{'})$ mod $m_i=0$ ……………………..… (13)

The following operation can be used to obtain the additive inverse $r_i^{'}$

$r_i^{'}=(m_i-r_i)$ mod m_i…………………...... (14)

One implementation of this property is subtraction, where subtraction is assumed to be the substitution of the opposite additive.

Multiplicative Inverse

The congruence of the multiplicative inverse r_i^{-1} of the residue r^i is defined by

$(r_i \text{ X } r_i^{-1})$ mod $m_i=1$ …………………… (15)

Where r_i^{-1} exists only if r_i and r_i^{-1} are relatively prime

A. Conversion between Conventional representation and RNS representation

To make use of the RNS's features and carry out processing in the residue domain, we need to be able to easily switch between the conventional (binary or analogue) and RNS representations. The process of changing a conventional representation into an RNS representation is known as forward translation (or forward conversion). By conceptually dividing all moduli by the specified conventional number and then working out the remaining divisions, this procedure can be executed. The simplest method to refer to any large collection of moduli is in this way. We demonstrate how this strategy might be further reduced for some particular module sets in this section, though. Since moving the digits to the right is comparable to dividing, the calculation is simplified.

The procedure of changing from RNS to conventional representations is referred to as "reverse conversion." In terms of speed and complexity, the reverse conversion process is more complicated and time-consuming. The Mixed-Radix Conversion (MRC) or Chinese Remainder Theorem is the foundation for the reverse conversion techniques (CRT). Parallelism is made possible during the execution of the conversion method by employing the CRT. The MRC is a sequential approach by nature. In general, employing VLSI to construct a reverse converter is expensive and complicated. [24-28].

2) Binary to RNS Representation Forward Conversion

The forward conversion step is crucial since it is seen as an overhead in the entire RNS. The moduli-set being used has a significant impact on the best technique. By the moduli used, forward converters are frequently divided into two kinds. Forward converters based on arbitrary moduli sets fall into the first type. Look-up tables are frequently used when creating these converters. Forward converters based on customised module sets make up the second group. Methods and structures for forward conversion are made simpler by using particular moduli-sets. Typically, pure combinational logic is used to create unique moduli-set converters.

In this part, we present numerous forward conversion architectures from binary to RNS form. On the basis of random moduli sets, we also suggest forward converters. Then, based on the unique moduli-set, we offer forward conversions {2n-1, 2n, 2n+1}. We show how to reduce design complexity overall to remove the burden the forward converter places on the system.

2.1) Arbitrary Moduli-Set Forward Converters

Forward converters built on specialised moduli sets are typically the most potent converters now on the market. On the other hand, some systems require a very wide dynamic range, which cannot be easily achieved through the use of specialised modules. Large complex ranges can be represented using one of two main techniques. The creation of appropriate algorithms and schemes is the initial stage in the development of arbitrary moduli-set forward converters. The second strategy is to create brand-new, distinctive module sets with lots of moduli in order to accurately reflect the wide dynamic spectrum. This method's unique five-moduli set, 2n-1, 2n, 2n+1, 2n-1-1, 2n+1-1} and its conversion circuits were proposed in [28].

In order to obtain the residues of all necessary powers of two with respect to modulus m, we employ the mathematical principle of addition to find the residue of a binary integer X with respect to a certain module m. If X is a binary integer, then the following is illustrated:

$\boldsymbol{X = x_{n-1}\, x_{n-2} \dots\dots\dots .\, x_1 x_0 \;=\; \sum_{j=0}^{n-1} x_j\, 2^j}$……..... (16)

The residue of X represented as:

$|\mathrm{x}|_{\mathrm{m}} = |\sum_{j=0}^{n-1} x_j \; 2^j\,|_{\mathrm{m}}$ .... (17)

Using the equation (1.10), we can write:

$|\mathrm{x}|_{\mathrm{m}} = |\sum_{j=0}^{n-1} |x_j \; 2^j|_m|_{\mathrm{m}}$ ....… (18)

Where x_j is either 0 or 1.

2.2) Special Moduli-Set Forward Converters

The suggested strategy for promoting and accelerating conversion processes is to select a unique moduli-set. The most potent practical converters in terms of speed, area, and power are special moduli set forward converters. Low moduli sets are those that fit this description. The particular moduli set utilized in this segment will be based on the most widely used moduli-set, {2n-1, 2n, 2n+1}. Two additional moduli sets {2n-1, 2n, 2n+1, 2n+1-1} and {2 n1-1, 2 n1+1, 2 n2-1, 2 n2+1} are also taken into consideration for our study work. These three moduli sets will be treated further because they are conjugate moduli sets.

2.2.1: The Special Moduli-Set, 3 Moduli set = {2n-1, 2n, 2n+1}

An overall calculation [2] is introduced to convert weighted number, X to RNS portrayal about the special 3-moduli-set {2n-1, 2n, 2n+1} by utilizing the numerical ideas made sense of below. Now; let us consider ‘X’ as a decimal number as follows,

$X= x_{3n-1}x_{3n-2} \dots x_{2n}\; x_{2n-1}x_{2n-2} \dots\dots\; x_n x_{n-1} \dots\dots\; x_0$

Based on the ‘n’ value, partition X into three blocks, each with n bits: B1, B2, and B3, where you can represent these blocks as follows:

$B_1=\sum_{j=2n}^{3n-1} x_j \;\; 2^{j-2n}$ .... (19)

$B_2=\sum_{j=n}^{2n-1} x_j \;\; 2^{j-n}$ ....… (20)

$B_3=\sum_{j=0}^{n-1} x_j \;\; 2^{j}$ ....…................... (21)

Therefore, $X=B_1 2^{2n}+ B_2 2^n+ B_3$……............................ (22)

The residue r_2 **is simply the first n least essential bits**, which can be obtained by moving X by n bits correctly. Residues r_3 and r_1 after simplification, written as follows,

$r_1=|B_1+B_2+B_3|_{2^n-1}$………………... (23)

r_2=First ‘n’ least significant bits..... (24)

$r_3=|B_1-B_2+B_3|_{2^n+1}$…………………. (25)

2.2.2: 4 Moduli set = {2^n-1, 2^n, 2^n+1, 2^{n+1}-1}

For 4-moduli-set {2^n-1, 2^n, 2^n+1, 2^{n+1}-1},

Based on ‘n’ value, X can be partition into blocks,

Residues r_4, r_3, r_2 and r_1 after simplification are written as follows

$r_1=|B_1+B_2+B_3+ B_4+ B_5|_{2^n-1}$............ (26)

r_2=First ‘n’ least significant bits.... (27)

$r_3=|B_1-B_2+B_3- B_4+ B_5|_{2^n-1}$.............. (28)

$r_4=|B_1+B_2+B_3+ B_4+ B_5|_{2^{n+1}-1}$…....... (29)

In 'r_4' calculations, based on '2^{n+1}-1' bits will be divided.

2.2.3: Conjugate Moduli set = {2^{n_1}-1, 2^{n_1}+1, 2^{n_2}-1, 2^{n_2}+1}

Conjugate moduli selection with n_1&n_2,
i.e. 4-moduli set with conjugate moduli set,
Moduli-set, m= {2^{n_1}-1, 2^{n_1}+1, 2^{n_2}-1, 2^{n_2}+1},
Based on 'n' value, X can be partition into blocks,
Residues R_4, R_3, R_0, and R_1 after simplification are written as follows

$r_1=|B_1+B_2+B_3+B_4+B_5|_{2^n-1}$………... (30)
r_2=First 'n' least significant bits…. (31)
$r_3=|B_1-B_2+B_3-B_4+B_5|_{2^n-1}$…….......... (32)
$r_4=|B_1+B_2+B_3+B_4+B_5|_{2^{n+1}-1}$……........(33)

Binary 'X' can be divided into blocks with '3' bits to accommodate
'2^{n_2}-1'=7 & '2^{n_2}+1'=9 values.

2.2.4) Theoretical calculations

2.2.4.1:3 Moduli set

Moduli-set=>m= {2^n-1, 2^n, 2^n+1} for n=3 => {m_1, m_2, m_3} = {7, 8, 9}
Decimal number=> X= $(167)_{10}$= $(10100111)_2$
Binary 'X' can be divided into blocks with '3' bits,
$B_3 = (111)_2 = (7)_{10}$, $B_2 = (100)_2 = (4)_{10}$, $B_3 = (010)_2 = (2)_{10}$,
From eqns. 23,24 & 25,
$r_1 = |B_1+B_2+B_3|_7 = |2+4+7|_7 = (6)_{10} == (110)_2$
r_2 = first 'n' least essential bits = $(7)_{10} = (111)_2$
$r_3 = |B_1-B_2+B_3|_7 = |2-4+7|_7 = (5)_{10} == (101)_2$
Residue set R= {r_1, r_2, r_3} = {6, 7, 5}

2.2.4.2:4 Moduli set

Moduli-set, m= {2^n-1, 2^n, 2^n+1, 2^{n+1}-1},
For n=2 => {m_1, m_2, m_3, m_4} = {3, 4, 5, 7}
Binary 'X' can be divided into blocks with '2' bits,
Decimal number, X= $(167)_{10}$= $(0010100111)_2$
$B_5 = (11)_2 = (3)_{10}$, $B_4 = (01)_2 = (1)_{10}$,
$B_3 = (10)_2 = (2)_{10}$,
$B_2 = (10)_2 = (2)_{10}$, $B_1 = (00)_2 = (0)_{10}$
From eqns. 30 to 33,
$r_1 = |B_1+B_2+B_3+B_4+B_5|_3$
$= |0+2+2+1+3|_3 = (2)_{10} = (10)_2$
r_2 = first 'n' least essential bits = $(3)_{10} = (11)_2$
$r_3 = |B_1 - B_2+B_3 - B_4 + B_5|_5$
$= |0-2+2-1+3|_5 = (2)_{10} = (10)_2$
Binary 'X' can be divided into blocks with '3' bits to accommodate
'2^{n+1}-1'=7 value,
$B_3 = (111)_2 = (7)_{10}$, $B_2 = (100)_2 = (4)_{10}$,
$B_1 = (010)_2 = (2)_{10}$,
$r_4 = |B_1+B_2+B_3|_7 = |2+4+7|_7 = (6)_{10} = (110)_2$
Residue set R= {r_1, r_2, r_3, r_4} = {2, 3, 2, 6}

2.2.4.3: Conjugate Moduli set

Moduli-set, m= {2^{n_1}-1, 2^{n_1}+1, 2^{n_2}-1, 2^{n_2}+1},
For n_1=1, n_2=3 => {m_1, m_2, m_3, m_4} = {1, 3, 7, 9}

Binary ‘X’ can be divided into blocks with ‘2’ bits,

Decimal number, $X= (167)_{10}= (0010100111)_2$

$B_5 = (11)_2 = (3)_{10}$, $B_4 = (01)_2= (1)_{10}$,

$B_3 = (10)_2 = (2)_{10}$,

$B_2= (10)_2 = (2)_{10}$, $B_1 = (00)_2= (0)_{10}$

From eqns. 30 to 33,

$r_1 = \left| B_1+B_2+B_3 + B_4 + B_5 \right|_1$

$= \left| 0+2+2+1+3 \right|_1 = (0)_{10} == (00)_2$

$r_2 = \left| B_1-B_2+B_3 - B_4 + B_5 \right|_5$

$= \left| 0-2+2-1+3 \right|_3 = (2)_{10} = (10)_2$

Binary ‘X’ can be divided into blocks with ‘3’ bits to accommodate ‘$2^n{}_2$-1’=7 & ‘$2^n{}_2$+1’=9 values,

$B_3 = (111)_2= (7)_{10}$, $B_2= (100)_2= (4)_{10}$, $B_1= (010)_2= (2)_{10}$,

$r_3 = \left| B_1+B_2+B_3 \right|_7 = \left| 2+4+7 \right|_7 = (6)_{10} == (110)_2$

$r_4 = \left| B_1-B_2+B_3 \right|_7 = \left| 2-4+7 \right|_7 = (5)_{10} == (101)_2$

Residue set $R= \{r_1, r_2, r_3, r_4\} = \{0, 2, 6, 5\}$

3) RNS to Binary Representation Reverse Conversion

With the aid of established reverse conversion techniques, such as the Chinese Remainder Theorem (CRT) or Mixed-Radix Conversion, reverse conversion can be accomplished (MRC). The MRC is a sequential approach by nature. CRT is also being utilized side by side. The requirement for a large module adder is the main drawback of the CRT-based R/B reverse converter, to sum up. This problem exists with every converter listed in the literature. Reverse conversion, one of the most challenging RNS procedures, has been a significant, if not the main, barrier to more widespread RNS implementation. R/B converter development on a VLSI platform is still generally expensive and challenging. In this section, the mathematical foundations of the CRT and MRC are established. Following this, potential reverse conversion applications of these techniques are described. Recently, reverse conversion has also been done using the New CRT-I and New CRT-II technologies.

a) Chinese Remainder Theorem (CRT)

The Chinese Remainder Theorem (CRT) is presented as follows in [1,2]; Given a system of pair-wise relatively prime moduli $\{m_1, m_2, m_3\ldots m_n\}$ and a representation of a number X's residues $\{r_1, r_2, r_3,\ldots, r_n\}$ in that system, where $r_i=|X|_{mi}$ links the number and its residues:

$$\left| X \right|_M = \left| \sum_{i=1}^{n} r_i \; \left| M_i^{-1} \right|_{mi} M_i \right|_M$$

$$X= \left| r_1 . \left| M_1^{-1} \right|_{m1} . M_1 + r_2 . \left| M_2^{-1} \right|_{m2} . M_2 + \ldots\ldots\ldots\ldots\ldots + r_n . \left| M_n^{-1} \right|_{mn} . M_n \right|_M \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (34)$$

Where ‘M’ is the product of the m_i’s, and $M_i=M/m_i$,

$\left| M_i^{-1} \right|_{mi}$ is multiplicative inverse with respect to m_i

The modular reduction on the left side can be skipped if the values involved are restricted so that the final value of X falls within the dynamic range.

3. a. 1) Theoretical calculations

3. a. 1.1:3 Moduli set

Moduli-set => $m= \{2^n-1, 2^n, 2^n+1\}$

For n=3 => $\{m_1, m_2, m_3\} = \{7, 8, 9\}$

Residue set => $R= \{r_1, r_2, r_3\} = \{6, 7, 5\}$

From eqn. 34,

$X= \left| r_1 . \left| M_1^{-1} \right|_{m1} . M_1 + r_2 . \left| M_2^{-1} \right|_{m2} . M_2 + r_3 . \left| M_3^{-1} \right|_{m3} . M_3 \right|_M$

$M = m_1.m_2.m_3 = 7.8.9=504$

$M_1 = M / m_1 = 8.9=72$

$M_2 = M / m_2 = 7.9=63$

$M_3 = M / m_3 = 7.8=56$

$X=\left| 6 . \left| 72^{-1} \right|_{7} . 72+ 7 . \left| 63^{-1} \right|_{8} . 63+ 5 . \left| 56^{-1} \right|_{9} . 56 \right|_{504}$

$X=\left| 13271 \right|_{504} = (167)_{10}$

Decimal number => $X= (167)_{10}= (10100111)_2$

3. a. 1.2:4 Moduli set

Moduli-set => $m= \{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$

For $n=2$ => $\{m_1, m_2, m_3, m_4\} = \{3, 4, 5, 7\}$

Residue set $R= \{r_1, r_2, r_3, r_4\} = \{2, 3, 2, 6\}$

From eqn. 34,

$X = \left| r_1 . \left| M_1^{-1} \right|_{m1} . M_1+ r_2 . \left| M_2^{-1} \right|_{m2} . M_2+ r_3 . \left| M_3^{-1} \right|_{m3} . M_3+ r_4 . \left| M_4^{-1} \right|_{m4} . M_4 \right|_{M}$

$M = m_1.m_2.m_3.m_4 = 3.4.5.7=420$

$M_1 = M / m_1 = 4.5.7=140$

$M_2 = M / m_2 = 3.5.7=105$

$M_3 = M / m_3 = 3.4.7=84$

$M_4 = M / m_4 = 3.4.5=60$

$X = \left| 20 . \left| 140^{-1} \right|_{3} .140+ 3 . \left| 105^{-1} \right|_{4} . 105+ 2 . \left| 84^{-1} \right|_{5} . 84+ 6 . \left| 60^{-1} \right|_{7} . 60 \right|_{420}$

$X= \left| 2267 \right|_{420} = (167)_{10}$

Decimal number => $X= (167)_{10}= (10100111)_2$

3. a. 1.3: Conjugate Moduli set

Moduli-set => $m= \{2^{n_1}-1, 2^{n_1}+1, 2^{n_2}-1, 2^{n_2}+1\}$

For $n_1=1, n_2=3$ => $\{m_1, m_2, m_3, m_4\} = \{1, 3, 7, 9\}$

Residue set => $R= \{r_1, r_2, r_3, r_4\} = \{0, 2, 6, 5\}$

From eqn. 34,

$X = \left| r_1 . \left| M_1^{-1} \right|_{m1} . M_1+ r_2 . \left| M_2^{-1} \right|_{m2} . M_2+ r_3 . \left| M_3^{-1} \right|_{m3} . M_3+ r_4 . \left| M_4^{-1} \right|_{m4} . M_4 \right|_{M}$

$M = m_1.m_2.m_3.m_4 = 1.3.7.9=189$

$M_1 = M / m_1 = 3.7.9=189$

$M_2 = M / m_2 = 1.7.9=63$

$M_3 = M / m_3 = 1.3.9=27$

$M_4 = M / m_4 = 1.3.7=21$

$X = \left| 0 . \left| 189^{-1} \right|_{1} .189+ 2 . \left| 63^{-1} \right|_{3} . 63+ 6 . \left| 27^{-1} \right|_{7} . 27+ 5 . \left| 21^{-1} \right|_{9} . 21 \right|_{189}$

$X= \left| 1301 \right|_{189} = (167)_{10}$

Decimal number => $X= (167)_{10}= (10100111)_2$

b) Mixed-Radix Conversion

Given a set of pair-wise relatively prime moduli {m1, m2, m3… mn} and a residue representation {r1, r2, r3… rn} in that system of some number X, that is ri=|X|mi, that number X can be uniquely represented in mixed-radix form as [2,10].

$X=\{z_1,z_2,z_3,\ldots..z_n\}$

Where $X= z_1+z_2m_1+z_3m_2 \quad m_1+\ldots\ldots..+ \quad z_nm_{n-1} \quad \ldots.m_1$ and $0\le z_i\le r_i$ ….. (35)

$z_1=r_1$

$z_2=\left| (r_2-z_1) \left| m_1^{-1} \right|_{m2} \right|_{m2}$

$z_3=\left| ((r_3-z_1) \left| m_1^{-1} \right|_{m3} -z_2) \left| m_2^{-1} \right|_{m3} \right|_{m3}$

$z_4=\left| (((r_4-z_1) \left| m_1^{-1} \right|_{m4} -z_2) \left| m_2^{-1} \right|_{m4} -z_3) \left| m_3^{-1} \right|_{m4} \right|_{m4}$

In general:

$z_n=\left| (((r_n-z_1) \left| m_1^{-1} \right|_{mn} -z_2) \left| m_2^{-1} \right|_{mn\ldots\ldots..} \ldots\ldots.z_{n-1}) \left| m_{n-1}^{-1} \right|_{mn} \right|_{mn}$

Where $\left|m_i^{-1}\right|_{mj}$ – Multiplicative inverse of m_i modulus m_j

A weighted, positional mixed-radix system is associated with an unweighted, non-positional RNS by the Mixed-Radix Conversion (MRC). To do the reverse conversion, only the values zi need to be obtained.

3. b. 1) Theoretical calculations

3. b. 1.1:3 Moduli set

Moduli-set => $m=\{2^n-1, 2^n, 2^n+1\}$

For n=3, Moduli set => $\{m_1, m_2, m_3\} = \{7, 8, 9\}$

Residue set => $R=\{r_1, r_2, r_3\} = \{6, 7, 5\}$

From eqn. 35,

$X= z_1+z_2m_1+z_3m_2\, m_1$

$z_1=r_1=6$

$z_2=\left|(r_2-z_1)\left|m_1^{-1}\right|_{m2}\right|_{m2}$

$=\left|(7-6)\left|7^{-1}\right|_{8}\right|_{8}=7$

$z_2=7$

$z_3=\left|((r_3-z_1)\left|m_1^{-1}\right|_{m3}-z_2)\left|m_2^{-1}\right|_{m3}\right|_{m3}$

$=\left|((5-6)\left|7^{-1}\right|_{9}-7)\left|8^{-1}\right|_{9}\right|_{9}=2$

$z_3=2$

$X= 6+7.7+2.8.7$

$X= (167)_{10}$

Decimal number => $X= (167)_{10}= (10100111)_2$

3. b. 1.2:4 Moduli set

Moduli-set => $m=\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$

For n=2 => $\{m_1, m_2, m_3, m_4\} = \{3, 4, 5, 7\}$

Residue set $R=\{r_1, r_2, r_3, r_4\} = \{2, 3, 2, 6\}$

From eqn. 35,

$X = z_1+z_2m_1+z_3m_2\, m_1+z_4m_3m_2m_1$

$z_1=r_1=2$

$z_2=\left|(r_2-z_1)\left|m_1^{-1}\right|_{m2}\right|_{m2}$

$=\left|(3-2)\left|3^{-1}\right|_{4}\right|_{4}=3$

$z_2=3$

$z_3=\left|((r_3-z_1)\left|m_1^{-1}\right|_{m3}-z_2)\left|m_2^{-1}\right|_{m3}\right|_{m3}$

$=\left|((2-2)\left|3^{-1}\right|_{5}-3)\left|4^{-1}\right|_{5}\right|_{5}=3$

$z_3=3$

$z_4=\left|(((r_4-z_1)\left|m_1^{-1}\right|_{m4}-z_2)\left|m_2^{-1}\right|_{m4}-z_3)\left|m_3^{-1}\right|_{m4}\right|_{m4}$

$=\left|(((6-2)\left|3^{-1}\right|_{7}-3)\left|4^{-1}\right|_{7}-3)\left|5^{-1}\right|_{7}\right|_{7}$

$z_4=2$

$X=2+3.3+3.3.4+2.3.4.5$

$=167$

$X= (167)_{10}$

Decimal number => $X= (167)_{10}= (10100111)_2$

3. b. 1.3: Conjugate Moduli set

Moduli-set => $m=\{2^{n_1}-1, 2^{n_1}+1, 2^{n_2}-1, 2^{n_2}+1\}$

For $n_1=1$, $n_2=3$ => $\{m_1, m_2, m_3, m_4\} = \{1, 3, 7, 9\}$

Residue set => $R=\{r_1, r_2, r_3, r_4\} = \{0, 2, 6, 5\}$

From eqn. 35,

$X = z_1+z_2m_1+z_3m_2\, m_1+z_4m_3m_2m_1$

$z_1=r_1=0$

$z_2=\left|(r_2-z_1)\left|m_1^{-1}\right|_{m2}\right|_{m2}$

$=\left|(2-0)\left|1^{-1}\right|_3\right|_3=2$

$z_2=2$

$z_3=\left|((r_3-z_1)\left|m_1^{-1}\right|_{m3}-z_2)\left|m_2^{-1}\right|_{m3}\right|_{m3}$

$=\left|((6-0)\left|1^{-1}\right|_7-2)\left|3^{-1}\right|_7\right|_7=6$

$z_3=6$

$z_4=\left|(((r_4-z_1)\left|m_1^{-1}\right|_{m4}-z_2)\left|m_2^{-1}\right|_{m4}-z_3)\left|m_3^{-1}\right|_{m4}\right|_{m4}$

$=\left|(((5-0)\left|1^{-1}\right|_9-2)\left|3^{-1}\right|_9-6)\left|7^{-1}\right|_9\right|_9$

$z_4=147$

$X=0+2.1+6.1.3+7.1.3.7$

$=167$

$X=(167)_{10}$

Decimal number => $X=(167)_{10}=(10100111)_2$

c) New Chinese Remainder Theorem-I

A number X is uniquely represented in New Chinese Remainder Theorem-I as [10,28] given a collection of pair-wise relatively prime moduli $\{m_1, m_2, m_3,\ldots m_n\}$ and a residue representation $\{r_1, r_2, r_3, \ldots r_n\}$ in that system of some number (ri=|X|mi). New CRT-I [2,29] can calculate the weighted number in the manner described below:

$$X = r_1+m_1 \text{ x}\left|k_1 (r_2-r_1) + k_2 m_2 (r_3-r_2) + k_3 m_2m_3(r_4-r_3) +\ldots\ldots + k_{n-1}m_2m_3\ldots\ldots m_{n-1} (r_n-r_{n-1})\right|_{m2m3..,\ldots mn} \qquad \ldots\ldots (36)$$

Where,

$$\left|k_1 x m_1\right|_{m2m3m4\ldots.mn} =1$$
$$\left|k_2 x m_1 x m_2\right|_{m3m4\ldots.mn} =1$$
$$\ldots\ldots$$
$$\left|k_{n-1} x m_1 x m_2 x m_3\ldots.. x m_{n-1}\right|_{mn} =1 \qquad \ldots\ldots\ldots\ldots (37)$$

Compared to the traditional CRT, the size of the final modulo viper decreased with the New CRT-I. In specifically, the New CRT-I can only be operated by a multi-operand modulus adder if the primary modulus of the moduli set is selected in the design 2k and the duplication of various moduli is in the design 2k-1. This type of architecture is described in [29].

3. c. 1) Theoretical calculations

3. c. 1.1:3 Moduli set

Moduli-set => $m= \{2^n-1, 2^n, 2^n+1\}$

For n=3 => $\{m_1, m_2, m_3\} = \{7, 8, 9\}$

Residue set => $R= \{r_1, r_2, r_3\} = \{6, 7, 5\}$

From eqn. 36 & 37,

$X= r_1+m_1 x\left|k_1 (r_2-r_1) +k_2m_2 (r_3-r_2)\right|_{m2m3}$

Where,

$\left|k_1m_1\right|_{m2m3}=1 \;\; => \left|k_1.7\right|_{8.9}=1$

$k_1=31$

$\left|k_2 x m_1 x m_2\right|_{m3} =1 => \left|k_2 x7x8\right|_9 =1$

$k_2=5$

$X= r_1+m_1 x\left|k_1 (r_2-r_1) +k_2m_2 (r_3-r_2)\right|_{m2m3}$

$= 6+7x\left|31 (7-6) +5.8 (5-7)\right|_{8.9}$

$=167$

$X=(167)_{10}$

Decimal number => $X=(167)_{10}=(10100111)_2$

3. c. 1.2:4 Moduli set

Moduli-set => $m= \{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$

For $n=2 \Rightarrow \{m_1, m_2, m_3, m_4\} = \{3, 4, 5, 7\}$

Residue set $R= \{r_1, r_2, r_3, r_4\} = \{2, 3, 2, 6\}$

From eqn. 36 & 37,

$X= r_1+m_1 x \left| k_1 (r_2-r_1) +k_2m_2 (r_3-r_2) +k_3m_2m_3(r_4-r_3) \right|_{m_2m_3m_4}$

Where,

$\left| k_1m_1 \right|_{m_2m_3m_4}=1 \Rightarrow \left| k_1.3 \right|_{4.5.7}=1$

$k_1=47$

$\left| k_2xm_1xm_2 \right|_{m_3m_4} =1 \Rightarrow \left| k_2x3x4 \right|_{5.7} =1$

$k_2=3$

$\left| k_2xm_1xm_2xm_3 \right|_{m_4} =1 \Rightarrow \left| k_3x3x4x5 \right|_{7} =1$

$k_3=2$

$X= r_1+m_1 x \left| k_1 (r_2-r_1) +k_2m_2 (r_3-r_2) +k_3m_2m_3(r_4-r_3) \right|_{m_2m_3m_4}$

$= 2+3x \left| 47 (3-2) +3.4 (2-3) +2.4.5(6-2) \right|_{4.5.7}$

$=167$

$X= (167)_{10}$

Decimal number $\Rightarrow X= (167)_{10}= (10100111)_2$

3. c. 1.3: Conjugate Moduli set

Moduli-set $\Rightarrow$ $m= \{2^{n_1}-1, 2^{n_1}+1, 2^{n_2}-1, 2^{n_2}+1\}$

For $n_1=1, n_2=3 \Rightarrow \{m_1, m_2, m_3, m_4\} = \{1, 3, 7, 9\}$

Residue set $\Rightarrow$ $R= \{r_1, r_2, r_3, r_4\} = \{0, 2, 6, 5\}$

From eqn. 36 & 37,

$X= r_1+m_1 x \left| k_1 (r_2-r_1) +k_2m_2 (r_3-r_2) +k_3m_2m_3(r_4-r_3) \right|_{m_2m_3m_4}$

Where,

$\left| k_1m_1 \right|_{m_2m_3m_4}=1 \Rightarrow \left| k_1.1 \right|_{3.7.9}=1$

$k_1=189$

$\left| k_2xm_1xm_2 \right|_{m_3m_4} =1 \Rightarrow \left| k_2x1x3 \right|_{7.9} =1$

$k_2=2$

$\left| k_2xm_1xm_2xm_3 \right|_{m_4} =1 \Rightarrow \left| k_3x1x3x7 \right|_{9} =1$

$k_3=11$

$X= r_1+m_1 x \left| k_1 (r_2-r_1) +k_2m_2 (r_3-r_2) +k_3m_2m_3(r_4-r_3) \right|_{m_2m_3m_4}$

$= 0+1x \left| 189 (2-0) +2.3 (6-2) +11.3.7(5-6) \right|_{3.7.9}$

$=167$

$X= (167)_{10}$

Decimal number $\Rightarrow X= (167)_{10}= (10100111)_2$

d) New Chinese Remainder Theorem-II

In the New Chinese Remainder Theorem-II, a number X is uniquely represented as [10,28] given a set of pair-wise relatively prime moduli $\{m_1, m_2, m_3\ldots m_n\}$ and a residue representation $\{r_1, r_2, r_3,\ldots, r_n\}$ in that system.

The following procedure, based on New CRT-II, determines the weighted number X where m1 m2 m3... mn [2], [29]:

```
Algorithm: translate ($(r_1,r_2\ldots r_v),(m_1, m_2, m_3,\ldots m_v),X$)
1) if $n > 2$, let $v= \lfloor n/2 \rfloor$, then
        translate ($(r_1,r_2\ldots r_n),(m_1, m_2, m_3,\ldots m_n),N_1$),
                    $M1=m_1m_2\ldots m_v$,
        translate ($(r_{v+1},\ldots r_n),( m_{v+1},\ldots m_n),N_2$),
                    $M2=m_{v+1}\ldots\ldots m_n$,
            findno ($N_1,N_2,M_1,M_2,X$).
2) if n=2, then find no($r_1,r_2,m_1,m_2,X$).
```

3) if n=1, then $X=\left|x1\right|_{m1}$.
Procedure: findno (r_1,r_2,m_1,m_2,X)
1) find a k0 such that $\left|k_0m_2\right|_{m1}=1$,
2) $X= r_2 + m_2\left|k_0\ (r_1-r_2)\right|_{m1.}$

It is clear that New CRT-overall II's algorithm has a tree-like architecture and reduces the size of the final modulus adder more than New CRT-I. The following equations [29] can be used to get the number X from its associated residues $\{r_1, r_2, r_3, r_4\}$ based on this technique for the four moduli set $\{m_1, m_2, m_3, m_4\}$.

$Y= r_3+m_3\left|k_3(x_4-x_3)\right|_{m4}$
$Z= r_1+m_1\left|k_2(x_2-x_1)\right|_{m2}$
$X=Z+m_1m_2\left|k_1(Y-Z)\right|_{m3m4}$ ………….. (38)

Where,

$\left|k_1xm_1xm_2\right|_{m3m4}=1$
$\left|k_2xm_1\right|_{m2}=1$
$\left|k_3xm_3\right|_{m4}=1.$ ….……….. (39)

k_1,k_2 and k_3 are the multiplicative inverses.

3. d. 1) Theoretical calculations

3. d. 1.1:3 Moduli set

Moduli-set => $m= \{2^n-1, 2^n, 2^n+1\}$
For n=3 => $\{m_1, m_2, m_3\} = \{7, 8, 9\}$
Residue set => $R= \{r_1, r_2, r_3\} = \{6, 7, 5\}$

For 3 moduli set,
Eqns. 38 & 39 can be modified,

$X=Z+m_1m_2\left|k_1(Y-Z)\right|_{m3}$
$Z= r_1+m_1\left|k_2(r_2-r_1)\right|_{m2}$
$Y= r_3$

Where,

$\left|k_1xm_1xm_2\right|_{m3m4}=1 => \left|k_1x7x8\right|_{9}=1$
$k_1=5$
$\left|k_2xm_1\right|_{m2}=1 => \left|k_2x7\right|_{8}=1$
$k_2=7$
$Y= r_3=5$
$Z= r_1+m_1\left|k_2\ .(r_2-r_1)\right|_{m2}= 6+7\left|7\ (7-6)\right|_{8}= 55$
$X=Z+m_1m_2\left|k_1(Y-Z)\right|_{m3}= 55+7.8\left|5(5-55)\right|_{9}$
$X= (167)_{10}$
Decimal number => $X= (167)_{10}= (10100111)_2$

3. d. 1.2:4 Moduli set

Moduli-set => $m= \{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$
For n=2 => $\{m_1, m_2, m_3, m_4\} = \{3, 4, 5, 7\}$
Residue set $R= \{r_1, r_2, r_3, r_4\} = \{2, 3, 2, 6\}$

From eqns. 38 & 39,

$Y= r_3+m_3\left|k_3(x_4-x_3)\right|_{m4}$
$Z= r_1+m_1\left|k_2(x_2-x_1)\right|_{m2}$
$X=Z+m_1m_2\left|k_1(Y-Z)\right|_{m3m4}$

Where,

$\left|k_1xm_1xm_2\right|_{m3m4}=1 => \left|k_1x3x4\right|_{5.7}=1$
$k_1=3$
$\left|k_2xm_1\right|_{m2}=1 => \left|k_2x3\right|_{4}=1$
$k_2= 3$

$|k_3 x m_3|_{m4} = 1 \Rightarrow |k_3 x 5|_3 = 1$

$k_3 = 2$

$Y = r_3 + m_3 |k_3(x_4 - x_3)|_{m4}$

$= 2 + 5. |2.(6-2)|_5$

$Y = 17$

$Z = r_1 + m_1 |k_2(x_2 - x_1)|_{m2}$

$= 2 + 3. |3.(3-2)|_4$

$Z = 11$

$X = Z + m_1 m_2 |k_1(Y-Z)|_{m3m4}$

$= 11 + 3.4 |3.(17-11)|_{5.7}$

$= 167$

$X = (167)_{10}$

Decimal number $\Rightarrow X = (167)_{10} = (10100111)_2$

3. d. 1.3: Conjugate Moduli set

Moduli-set $\Rightarrow$ $m = \{2^{n_1}-1, 2^{n_1}+1, 2^{n_2}-1, 2^{n_2}+1\}$

For $n_1=1, n_2=3 \Rightarrow \{m_1, m_2, m_3, m_4\} = \{1, 3, 7, 9\}$

Residue set $\Rightarrow$ $R = \{r_1, r_2, r_3, r_4\} = \{0, 2, 6, 5\}$

From eqns. 38 & 39,

$Y = r_3 + m_3 |k_3(x_4 - x_3)|_{m4}$

$Z = r_1 + m_1 |k_2(x_2 - x_1)|_{m2}$

$X = Z + m_1 m_2 |k_1(Y-Z)|_{m3m4}$

Where,

$|k_1 x m_1 x m_2|_{m3m4} = 1 \Rightarrow |k_1 x 1 x 3|_{7.9} = 1$

$k_1 = 18$

$|k_2 x m_1|_{m2} = 1 \Rightarrow |k_2 x 1|_3 = 1$

$k_2 = 4$

$|k_3 x m_3|_{m4} = 1 \Rightarrow |k_3 x 7|_1 = 1$

$k_3 = 8$

$Y = r_3 + m_3 |k_3(x_4 - x_3)|_{m4}$

$= 6 + 7. |8.(5-6)|_9$

$Y = 8$

$Z = r_1 + m_1 |k_2(x_2 - x_1)|_{m2}$

$= 0 + 1. |4.(2-0)|_3$

$Z = 5$

$X = Z + m_1 m_2 |k_1(Y-Z)|_{m3m4}$

$= 5 + 1.3. |18.(8-5)|_{7.9} = 167$

$X = (167)_{10}$

Decimal number $\Rightarrow X = (167)_{10} = (10100111)_2$

C. Operations in the Residue Number System

The main advantage of the residue number system (RNS) is the lack of carryover in addition and multiplication between columns. A closed calculation is one that is completed totally inside each residue position. As a result, since long numbers are not completely determined by the most prominent modulus location, expansion and increase operations can be performed on them at a rate comparable to that of short numbers. Recall that an operation on a lengthy word takes longer in the traditional linear weighted number system because carry propagation is involved.

With previously mentioned advantage, RNS has more advantages in activities like expansion, deduction, augmentation, recursive augmentations with duplications

(Augmentation and Gathering (Macintosh)) tasks with less time and shortcoming lenient. This; will be used in sign detection, division, and overflow-like operations. From [30], RNS arithmetic operations are calculated in the following manner.

Examples of additions in moduli set m= {7, 8, 9},

Residue arithmetic are:

i) Addition operation

$$\begin{array}{ccc} & \longrightarrow & \{6,7,5\} \\ (167)_{10} & & \\ + & \longrightarrow & +\{6,7,5\} \\ (167)_{10} & & \\ \hline & \longrightarrow & \{5,6,1\} \\ 334)_{10} & & \end{array}$$

Decimal equivalent => $(167)_{10} + (167)_{10} = (334)_{10}$

RNS equivalent for moduli set {7, 8, 9} =>

{6, 7, 5} + {6, 7, 5} = {5, 6, 1}

ii) Subtraction operation

$$\begin{array}{ccc} & \longrightarrow & \{6,7,5\} \\ (167)_{10} & & \\ - & \longrightarrow & -\{6,7,5\} \\ (167)_{10} & & \\ \hline & \longrightarrow & \{0,0,0\} \\ (000)_{10} & & \end{array}$$

Decimal equivalent => (167)10 + (167)10 = (000)10

RNS equivalent for moduli set {7, 8, 9} =>

{6, 7, 5} - {6, 7, 5} = {0, 0, 0}

iii) Multiplication operation

$$\begin{array}{ccc} (167)_{10} & \longrightarrow & \{6,7,5\} \\ \text{x } (167)_{10} & \longrightarrow & \text{x}\{6,7,5\} \\ \hline & \longrightarrow & \{1,1,7\} \\ 27889)_{10} & & \end{array}$$

Decimal equivalent => (167)10 x (167)10 = (27889)10

RNS equivalent for moduli set {7, 8, 9} =>

{6, 7, 5} x {6, 7, 5} = {1, 1, 7}

D. Applications of Residue Number System

From the reacquire of RNS, it is applied to regions in which basic number juggling activities like augmentations and multiplications [31] for example, Computerized Signal Handling, for example, advanced separating, convolution, Quick Fourier change, advanced picture handling, Low power plan, cryptography, bioinformatics and so forth. This; part surveys various regions and designs where RNS has been applied.

RNS has been used in applications requiring complicated number arithmetic, fault-tolerant systems, and high-speed arithmetic because of its advantages. RNS has increasingly been applied to AI, blockchain technology, cloud storage, and network routing. The numerous

RNS-based applications are compiled in Table I with clear areas of commitment.

Table I: RNS Applications with explicit areas of commitments

Paper	Application Field	Particular Field of Application	Challenges
[R32]	Artificial Intelligence	• Increasing the effectiveness of the base extension by employing a neural network with orthogonal bases and CRT as its foundation. • Creation of a CNN architecture utilizing RNS to reduce resource costs • To make convolutional neural networks less computationally complex, RNS was combined with the Winograd method (CNN) • RNS was utilized to improve the Multiply-and-Accumulate (MAC) procedure used during network assessment. RNSnet maps straightforward neural network operations for more memory and data access.	• Improving a neural network's overall inference by enhancing the basic multiplication and accumulating operations. Enhancing neurons' non-linearity to enhance an artificial neural network's performance (ANN). • Deep neural networks' energy efficiency and memory capacity • Convolutional Neural Networks' resource-intensive and complex operations (CNN)
[R33]	Digital Image and Video Processing	• RNS approach for spatial and frequency domain digital image filtering to enhance integration circuit performance	• Issues with digital image processing speed, security, and power consumption
[R34]	Digital Signal Processing	• Quadratic residue number system (QRNS) will be used to expedite the speed of complex arithmetic. • To save logic resources, a low-cost fault-tolerant finite impulse response (FIR) filter was created using RNS. • Quadratic residue number system (QRNS) will be used to expedite the speed of complex arithmetic.	• Enhancing FIR filter performance

[R35]	Digital Watermarking	• Inverse CRT is used to apply CRT after choosing two co-prime numbers in order to determine where the embedding is located. • Some pixels get a superfluous bit added as a watermark, while the remainder become residues.	The robustness of watermarking systems should be improved, as well as the reversible and fragile characteristics of watermarking.
[R36]	Bioinformatics	• Carry-free propagation characteristics and parallelism On an FPGA, RNS was employed to speed up the algorithm.	• enhancing the Smith-Waterman algorithm's speed.
[R37]	Block Chain and IoT	• RN-based Elliptic Curve Cryptography Cipher on ensuring an appropriate level of security • Applied to improving the storage mechanism without changing the blockchain's architecture; • Some pixels get a superfluous bit added as a watermark, while the remainder become residues.	• Improving IoT security; Blockchain development's inefficient storage of a huge number of data;
[R38]	Cryptography	• A fault injection side channel attack mitigation scheme based on elliptic curve cryptography. • Data in RNS format and RNS circuits are used to execute point multiplication. • Enhance the image scrambling technique using a set of three moduli to get a significantly less encrypted image size. • The Redundant Residue Number System (RRNS), which uses a fresh approach to error-correcting codes and top-secret sharing protocols. • Using RNS to make the entire video encryption process less computationally complex. • A fault injection side channel attack mitigation scheme based on elliptic curve cryptography.	• The difficulty of computation used for the entire video encryption. • Aside channel fault injection attack on an elliptic curve cryptosystem. • Ineffective algorithm for image encoding

[R39]	Data Communication and Networking	• Communication network fault detection • Energy efficiency and dependability in wireless sensor networks • Data transmission reliability in wireless sensor networks • Software Defined Networks' use of tabless routing	• In Wireless Sensor Networks, there is only sporadic and unstable transmission. The distributed self-diagnosis protocol algorithm's inability to run in parallel for fault detection in communication networks • High energy use and erratic WSN performance •Routing delay in software-defined networks
[R40]	Cloud Storage	• Carry-free propagation characteristics and parallelism On an FPGA, RNS was employed to speed up the algorithm. • RNS-based homomorphic encryption for cloud storage	• Independent cloud collaboration, which lowers security.

From the previously mentioned applications, one of the FIR channels uses of RNS [41] is considered and made sense of in an accompanying way.

D.1) FIR filters

D.1.1) Jenkins and Leon FIR filters

Jenkins and Leon's [Jenk77] FIR filter implementations make advantage of ROM-based multipliers, which perform well in [4]. A common example of their application is quite illuminating. Think about the first-order filter that the equation describes.

$y(n) = a_o.\ u(n) + a_1.u(n\text{-}l)$ .... (40)

Where, with inputs $u(n) = 30$ and $u(n\text{-}l) = 97$, $a_o = 127$ and $a_1 = -61$. It must be output as -2107. They settled on a moduli set of {19, 23, 29, 31} with a dynamic range of 392,863 and a word length of roughly 18 bits. It is clear that the coefficients in this RNS are $a_o = \{13\ 12, 11, 3\}$ and $a_1 = \{15\ 8, 26, 1\}$.

When $M_i = M_i/m_i$, the multiplicative inverses of M_i are calculated as follows:

$M_1 = M/m_1 = 20677$; $(1/M_1) \bmod m_1 = 4$
$M_2 = M/m_2 = 17081$; $(1/M_2) \bmod m_2 = 20$
$M_3 = M/m_3 = 13547$; $(1/M_3) \bmod m_3 = 22$
$M_4 = M/m_4 = 12673$; $(1/M_4) \bmod m_4 = 5$

The modified filter coefficients that are produced by multiplying the original filter coefficients by (l/Mi) mod mi are as follows:

$a_0' = \{14, 10, 10, 15\}$
$a_t' = \{3, 22, 21, 5\}$

Evidently, in order to obtain the terms a_0.u(n) and at.u(n-1) that, when added, produce the necessary result, the terms u(n) and u(n-1) must be multiplied by these. Aware of that,
u(n) = 30 = {11, 7, 1, 30}, u(n-1) = 97= {2, 5, 10,4}
Now,
y'(n)={8, 19, 17, 5} .... (41)
Evidently, to obtain the CRT result, this must be weighted by Mi and added.
A table that stores the $\sum y_{ij}.\ 2^{j-1}$ is necessary for the bit slice method.
Since y'_i are 5 bit words, M_i corresponds to all possible combinations of the jth bits of y_t', y_2', y_3', and y_4' where j = 0 to 4. As a result, five table look-ups followed by summing are required to determine the result since y'(n) is represented as a set of five bit words.
Categorially, the $y_1'(n)$, $y_2'(n)$, $y_3'(n)$, $y_4'(n)$ in binary form from eq.(41) are
$y_t'(n)$ =01000
$y_2'(n)$ =10011
$y_3'(n)$ =10001
$y_4'(n)$ = 00101.

Thus, the PROM is addressed by the MSB (i.e., j=4) word 0110 in order to retrieve 30628, which is then multiplied by two to produce 61256. The word that matches 0001 next reads as 20677, which will be added to 61256. The final result is still obtained using the earlier procedure.

LITERATURE SUMMARY

The data conversion, i.e. Forward Conversion, Reverse Conversion, and RNS arithmetic examples illustrated in the literature review, it give information that less carry overhead is carried out through calculations and design. It can imply that less hardware requirement and less computation time are necessary in RNS processor. In RNS-based system implementation, Forward conversion and RNS arithmetic absorb exact performance requirements in utmost applications. While; on the contrary, reverse converters differ in performance requirements. In this connection, in summary, reverse converters are barely considered for the selected conjugate moduli sets. From [3, 5], it can be concluded concerning the hardware requirements and delay arranged in Table II.

Table II: Area and delay Necessities of conjugate moduli set

Design	Moduli Set	Hardware requirements	Delay
[5]	$\{2^n-1, 2^n, 2^n+1\}$	$(6n+1)A_{FA}+(n+3)A_{AND/OR}$ $+(n+1)A_{XOR/XNOR}+2n\ A_{2:1MUX}$	$(n+2)\tau_{FA}$ $+\ \tau_{MUX}$
[5]	$\{2^n-1, 2^n, 2^n+1,2^{n+1}-1\}$	$(9n+5+((n-4)(n+1)/2))A_{FA}$ $+2nA_{EXNOR}+2nA_{OR}+(6n+1)A_{INV}$	$[(23n+12)/2]$ τ_{FA}
[3]	$\{2^{n_1}-1, 2^{n_1}+1,$ $2^{n_2}-1, 2^{n_2}+1\}$ For $X_i =\lvert X\rvert_{2ni-1}$ and $X_i =\lvert X\rvert_{2ni+1}$.	$12n_i+2$ (CSA-cost in number of FAs)	5 (Delay in number of levels of FAs)

Based on Table II information,
For n=3,

(I) for special moduli set $\{2^n-1, 2^n, 2^n+1\}$ =>

Hardware requirements =

$(6.2+1)\ A_{FA}+ (3+3)A_{AND/OR}+(3+1)A_{XOR/XNOR}+2.3\ A_{2:1MUX}$

$(13)\ A_{FA}+(6)A_{AND/OR}+(4)A_{XOR/XNOR}+6\ A_{2:1MUX}$

Delay =>

$(3+2)\tau_{FA}+ \tau_{MUX}$ => $(5)\tau_{FA}+ \tau_{MUX}$

(II) for moduli set $\{2^n-1, 2^n, 2^n+1,2^{n+1}-1\}$ =>

Hardware requirements =

$(9.3+5+((3-4)(3+1)/2))A_{FA} +2.3A_{EXNOR}+2.3A_{OR}+(6.3+1)A_{INV}$

$(30))A_{FA} +(6)A_{EXNOR}+(6)A_{OR}+(19)A_{INV}$

Delay =>

$[(23.3+12)/2]\ \tau_{FA}$ => $(41)\ \tau_{FA}$

(II) for moduli set $\{2^{n_1}-1, 2^{n_1}+1, 2^{n_2}-1, 2^{n_2}+1\}$ =>

Hardware requirements =

With n_i=1=>12.1+2=14 (CSA-cost in number of FAs)

With n_i=3=>12.3+2=38 (CSA-cost in number of FAs)

Delay =>

With n_i=1=>5(Delay in number of levels of FAs))

With n_i=3=>5(Delay in number of levels of FAs)

The calculations as mentioned above reveal that the conjugate moduli set yields better performance in hardware requirements and delay point of view.

CONCLUSION

The following points summarize the work's key components:

The RNS is a profoundly spurring number framework due to its better speed and low power utilization in applications requiring broad duplicate and-amass activities. DSP; applications are a outstanding possibility for such applications. The; decrease in power use is exceptionally reassuring for convenient clients.

We investigated methods and applications for forwarding data transfer, i.e. from conventional to RNS representations. The technique is used as a preprocessing step to encrypt the data in residue form concerning specified moduli. Numerous; methods and frameworks for converting data from RNS to standard representation are discussed. After; the RNS processor interprets the residue encoded data; the data must change back to their conventional format. This technique is considered to be a stage in the post-processing process. Reverse; conversion, which has a significant, if not the primary, an impediment to the widespread adoption of RNS, is one of the most challenging RNS processes. The; output data may be binary or analog. Execution of RNS can be further developed by founded on moduli set segment, information change, for example, from Customary to RNS, as well as the other way around and RNS math. In conjunction with, conjugate moduli sets selected are special moduli set {2n-1, 2n, 2n+1}, {2n-1, 2n, 2n+1, 2n+1-1} and conjugate moduli set {2 n1-1, 2 n1+1, 2 n2-1, 2 n2+1} for analyzing moduli set selection consequence on the performance of RNS; The RNS can be extended to [41]. From; the literature review and summary in [3,5], this is concluded that the selection of conjugate moduli has a greater impact on the RNS performance; these results can be a foremost choice in RNS-based applications.

References

- Taylor, "Residue Arithmetic A Tutorial with Examples," in Computer, vol. 17, no. 5, pp. 50-62, May 1984, doi: 10.1109/MC.1984.1659138.S. C. Yang, "Toward a wireless world," IEEE Technol. Soc. Mag., vol. 26, no. 2, pp. 32–42, Jun. 2007.
- K. Navi, A. S. Molahosseini and M. Esmaeildoust, "How to Teach Residue Number System to Computer Scientists and Engineers," in IEEE Transactions on Education, vol. 54, no. 1, pp. 156-163, Feb. 2011, doi: 10.1109/TE.2010.2048329.
- Skavantzos and M. Abdallah, "Implementation issues of the two-level residue number system with pairs of conjugate moduli," in IEEE Transactions on Signal Processing, vol. 47, no. 3, pp. 826-838, March 1999, doi: 10.1109/78.747787.
- Mohan, P.V.A. (2002). Applications of Residue Number Systems. In: Residue Number Systems. The Springer International Series in Engineering and Computer Science, vol 677. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-0997-4_8
- Ananda Mohan, P.V. (2016). RNS to Binary Conversion. In: Residue Number Systems. Birkhäuser, Cham. https://doi.org/10.1007/978-3-319-41385-3_5
- S. C. Yang, "Toward a wireless world," IEEE Technol. Soc. Mag., vol. 26, no. 2, pp. 32–42, Jun. 2007.
- R. Schneiderman, "DSPs evolving in consumer electronics applications," IEEE Signal Process. Mag., vol. 27, no. 3, pp. 6–10, May 2010.
- T. Stouraitis and V. Paliouras, "Considering the alternatives in lowpower design," IEEE Circuits Devices Mag., vol. 17, no. 4, pp. 23–29,Jul. 2001.
- H. L. Garner, "The residue number system," IRE Trans. Electron. Comput., vol. 8, no. 2, pp. 140–147, 1959.
- C.-H. Chang, A. S. Molahosseini, A. A. E. Zarandi, and T. F. Tay, "Residue number systems: A new paradigm to datapath optimization for low-power and high-performance digital signal processing applications," IEEE Circuits Syst. Mag., vol. 15, no. 4, pp. 26–44, 4th Quart. 2015.
- R. Conway and J. Nelson, "Improved RNS FIR Filter Architectures," IEEE Transactions On Circuits and Systems II, Vol. 51, No. 1, pp. 26-28, 2004
- W. Wei et al., "RNS application for digital image processing," Proceedings of the 4th IEEE international workshop on system-on-chip for real time applications, Canada, pp. 77-80, 2004.
- S. Yen, S. Kim, S. Lim and S. Moon, "RSA Speedup with Chinese Remainder Theorem Immune against Hardware Fault Cryptanalysis," IEEE Transactions on Computers, vol. 52, no. 4, pp. 461-472, 2003.
- J. Ramirez, et al., "Fast RNS FPL-Based Communications Receiver Design and Implementation," Proceedings of the 12th Int'l Conf. Field Programmable Logic, pp. 472-481, 2002.
- Omondi and B. Premkumar, "Residue Number System: Theory and Implementation," Imperial College Press 2007, ISBN 978-1-86094-866-4.
- G. C. Cardarilli, A. Nanareelli, and M. Re, "Reducing power dissipation in FIR filters

using the residue number system," Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, Vol. 2, pp. 320-323, 2000.

- Nanareelli, M. Re, and G. C. Cardarilli, "Tradeoffs between residue number system and traditional FIR filters," The 2001 IEEE International Symposium on Circuits and Systems, Vol. 2, pp. 305-308, 2001.
- W. K. Jenkins, "Finite Arithmetic Concepts in Handbook for DSP," S. K. Mitra and J. F. Kaiser, eds., Wiley, 1993, pp. 611-675.
- Ghassem Jaberipur and Seyed Hamed Fatemi Langroudi," (4 + 2 logn)ΔG Parallel Prefix Modulo Adder 2n-3 via Double Representation of Residues in [0,2], IEEE Transactions on Circuits and Systems II: Express Briefs,2015.
- Azadeh Alsadat Emrani Zarandi, Amir Sabbagh Molahosseini, Leonel Sousa, and Mehdi Hosseinzadeh," An Efficient Component for Designing Signed Reverse Converters for a Class of RNS Moduli Sets of Composite Form {2k, 2P − 1}",IEEE Transactions On Very Large Scale Integration (VLSI) Systems, PP. 1063-8210 , 2016.
- Zeinab Torabi and Ghassem Jaberipur," Low-Power/Cost RNS Comparison via Partitioning the Dynamic Range", IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, 2016.
- Piotr Patronik and Stanisław J. Piestrak," Hardware/Software Approach to Designing Low-Power RNS-Enhanced Arithmetic Units",IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS–I: REGULAR PAPERS, 2017.
- Omondi and B. Premkumar, "Residue Number System: Theory and Implementation," Imperial College Press 2007, ISBN 978-1-86094-866-4.
- Cao, C. Chang, and T. Sirkanthan, "A residue-to-binary converter for a new five-moduli set," IEEE Transactions on Circuits and Systems, 35 (11), 1998.
- G. C. Cardarilli, A. Nanareelli, and M. Re, "Reducing power dissipation in FIR filters using the residue number system," Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, Vol. 2, pp. 320-323, 2000.
- Nanareelli, M. Re, and G. C. Cardarilli, "Tradeoffs between residue number system and traditional FIR filters," The 2001 IEEE International Symposium on Circuits and Systems, Vol. 2, pp. 305-308, 2001.
- Parhami and C. Y. Hung, "Optimal Table Lookup Schemes for VLSI Implementation of Input/Output Conversions and other Residue Number Operations," In: VLSI Signal Processing VII, IEEE Press, New York, 1997.
- M. A. Soderstrand, W. K. Jenkins, G. A. Jullien and F. J. Taylor, "Residue Number System Arithmetic: Modern Applications in Digital Signal Processing," IEEE Press, New York, 1986.
- Yuke Wang, "Residue-to-binary converters based on new Chinese remainder theorems," in IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 47, no. 3, pp. 197-205, March 2000, doi: 10.1109/82.826745.
- Samavi, Shadrokh. (2014). Residue Number System (RNS).
- Afeez Adeshina Oke 1 , Babatunde Akinbowale Nathaniel 1 , Balogun Fatimah Bukola 1 , Oloyede Abdulkarim Ayopo," RESIDUE NUMBER SYSTEM BASED APPLICATIONS: A LITERATURE REVIEW" Anale. Seria Informatică. Vol. XIX fasc.

1 – 2021 Annals. Computer Science Series. 19 th Tome 1st Fasc. – 2021

- (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, Afzallikusha, & Pedram, 2019) (Zhi-Gang & Mattina, 2020
- Vassalos & Bakalis, 2013) (Kenneth & J., 1977) (Soudris, DSgouropoulos, Tatas, & Padidis, 2003) (Pontarelli, Cardarilli, Re, & Salsano, 2008) (Luan, Chen, Ge, & Wang, 2014
- 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
- Mei, Gao, Guo, Zhao, & Yang, 2019) (Pandey, Mitharwal, & Karmakar, 2019
- 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
- Wei Wang, Swamy, & Ahmad, 2004) (Toivonen, 2006) (Taleshmekaeil & Mousavi, 2010) (S. Alhassan, Gbolagade, 2013) (Nikolai I Chervyakov, Lyakhov, Nikolai, & Bogayevskiy, 2019
- Gomathisankaran, Tyagi, & Namuduri, 2011) (Kar, Sur, Basak, Sukla, & Das, 2016) (Tchernykh, Babenko, Chervyakov, & Mirandalópez, 2018
- 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
- Rajalakshmi & Nivedita, 2018) (Kehinde & Alagbe, 2018) (Mensah, Bankas, & Iddrisu, 2018
- 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)
- AyyavaruReddy, Y. and Sekhar, B. (2016) An Efficient Reverse Converter Design for Five Moduli Set RNS. International Journal of Advanced Research in Computer and Communication Engineering, 5, 208-212.