Scientific Research Publishing

Journal of Computer and Communications, 2021, 9, 57-66
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

New Reverse Conversion for Four-Moduli Set and Five-Moduli Set

Abdul-Mumin Salifu

Information Systems and Technology, C.K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana
Email: asalifu@cktutas.edu.gh

How to cite this paper: Salifu, A.-M. (2021) New Reverse Conversion for Four-Moduli Set and Five-Moduli Set. *Journal of Computer and Communications*, **9**, 57-66.
https://doi.org/10.4236/jcc.2021.94004

Received: February 28, 2021
Accepted: April 13, 2021
Published: April 16, 2021

Abstract

Most reverse conversions in Residue Number Systems (RNS) are based on the Chinese Remainder Theorem (CRT) and the Mixed Radix Conversion (MRC). The complexity of the circuitry of the CRT is high due to the large modulo-M operation. The MRC has a simple circuitry but it's a sequential process in nature. The purpose of this research is to obtain an efficient reverse conversion method to reduce the computational overhead found in the conventional reverse conversion algorithms. In this paper, new algorithms for reverse conversion in RNS for four-moduli set and five-moduli set have been proposed and their correctness evaluated. Numerical evaluations to ascertain the correctness and simplicity of the algorithm have been presented. These algorithms have fewer multiplicative index operations than those in the conventional CRT and MRC. The large modulo-M operation has been eliminated which reduces the computational overhead.

Keywords

Chinese Remainder Theorem, Forward Conversion,
Mixed Radix Conversion Residue Number Systems, Reverse Conversion

1. Introduction

In recent times, digital processors based on RNS contribute significantly in many digital signal processing applications. In these related applications, addition is the common operation that is carried out using different kinds of adders. In RNS based arithmetic operations, there is no carry propagation which is a great limiting factor of archiving high speed processing time. Digital processors built based on RNS offer the advantages of parallelism and fault tolerance [1] [2] [3] [4]. In RNS, Integers are represented by taking the modulus operation of the given integer over a set of moduli. The respective residues represent the integer in RNS. Some of the problems that limit the full utilization of RNS are difficult in carrying out division, magnitude comparison, sign detection and scaling [5] [6].

To utilize RNS based processors and their advantages, conventional number systems such as numbers in binary or decimal representation must be converted into residues and that is referred to as Forward Conversion, then the RNS processor does the arithmetic operations and results are converted back to their binary or decimal equivalent and the process is referred to as Reverse Conversion. The Forward Conversion process is very fast and straight forward. The Reverse Conversion is more difficult and introduces much overhead in terms of time complexity and area [7] [8].

The conversions component of RNS structure is very essential in determining the performance of RNS. The conversion process may be computational intensive in the circuitry and may introduce undue propagation delay and increase the area of the general architecture of the RNS system. This can derail the importance of using RNS in digital processor applications. To completely build an RNS processor that can replace the currently used digital processors, there is the need to build efficient conversion algorithms that will still make the gains in terms of speed in RNS a worthwhile adventure.

Currently available conversion algorithms are based on the Chinese Remainder Theorem or the Mixed Radix Conversion techniques. The MRC is inherently a sequential process in nature; in computing the mixed radix digits, the correctness of the subsequent mixed radix digits depends on the preceding mixed radix value. This is a major challenge as an error in one mixed radix value will lead to an error in the preceding mixed radix value. The major problem with the Chinese remainder theorem based reverse conversion techniques is the need of a large modulo adder in the last stage. This can derail the general performance of the RNS architecture because it increases the computational intensity of the conversion process.

In this paper, a new reverse conversion technique is presented for a four modular set and five modular set. The paper seeks to obtain a new reverse algorithm without the need of a large modulo adder found in the CRT conversion technique. The proposed conversion technique also seeks to resolve the inherent sequential process found in the MRC. This technique is based on what is proposed by Asiedu and Salifu (2021) but limited to a two-moduli set and a three-moduli set hence limiting the dynamic range (few numbers can be represented). The new scheme seeks to increase the dynamic range in order to allow applications with huge numbers to be represented.

2. Related Algorithms

Traditionally, there are two main known algorithms for reverse conversion, the Chinese Remainder Theorem (CRT) and the Mixed Radix Conversion (MRC). Other variants based on the CRT and MRC have been proposed. The advantage of the CRT is because the data conversion can be parallelized to limit inherent errors in the conversion process. The disadvantage is that it has slow modulo-M operation. MRC has less complex circuitry but by nature, it is a sequential process where the value of the subsequent mixed radix depends on the preceding mixed radix value. This implies that an error in one mixed radix value will lead to an error in the subsequent mixed radix value.

Asiedu and Salifu (2021) proposed a recent reverse conversion algorithm for two-moduli set and three-moduli set that are very simple and with fewer multiplicative inverse operations than there are in the traditional algorithms like the Chinese Remainder Theorem (CRT) and Mixed Radix Conversion (MRC).

2.1. Chinese Remainder Theorem

Given the moduli set $\{m_1, m_2, m_3, \cdots, m_n\}$ and the RNS representation of an integer X be represented as $\{x_1, x_2, x_3, \cdots, x_n\}$. Then the Chinese Remainder Theorem is as follows:

$$|X|_M = \left|\sum_{i=1}^{N} x_i \left|M_i^{-1}\right|_{m_i} M_i\right|_M \tag{2.1}$$

where M is the product of the m_i M_i^{-1} are the multiplicative inverse of M_i with respect to m_i $M_i = \frac{M}{m_i}$ [9] [10].

2.2. Mixed Radix Conversion

Consider the moduli set $\{m_1, m_2, m_3, \cdots, m_n\}$ and the RNS representation of an integer X be given as $\{x_1, x_2, x_3, \cdots, x_n\}$, it's decimal equivalent is computed based of the Mixed Radix Conversion as follows:

$$X = d_1 + d_2 m_1 + d_3 m_1 m_2 + \cdots + d_n m_1 m_2 m_3 \cdots m_{n-1} \tag{2.2}$$

where $d_i, i = 1, 2, \cdots, n$ are the Mixed Radix Digits (MRDs) and computed as follows:

$$d_1 = x_1$$

$$d_2 = \left|(x_2 - d_1)\left|m_1^{-1}\right|_{m_2}\right|_{m_2}$$

$$d_3 = \left|\left((x_3 - d_1)\left|m_1^{-1}\right|_{m_3} - d_2\right)\left|m_2^{-1}\right|_{m_3}\right|_{m_3}$$

$$\vdots$$

$$d_n = \left|\left(\cdots\left((x_3 - d_1)\left|m_1^{-1}\right|_{m_n} - d_2\right)\left|m_2^{-1}\right|_{m_n} - \cdots - d_{n-1}\right)\left|m_{n-1}^{-1}\right|_{m_n}\right|_{m_n} \quad [11] \tag{2.3}$$

That is, X in the interval $[0, M]$ can be unambiguously represented.

3. Proposed Algorithms

The proposed algorithm is based on the algorithm for reverse conversion proposed by Asiedu and Salifu (2021). The authors proposed new algorithms for reverse conversion for two-moduli set and for three-moduli set. In this paper, four-moduli set and a five-moduli set are proposed.

3.1. Algorithm for Four Moduli Set

Given a four moduli set $m=\{m_1,m_2,m_3,m_4\}$ and residues $r=(r_1,r_2,r_3,r_4)$, moduli set m and residues r can be represented in congruence form as:

$$X \equiv r_1 \bmod m_1 \tag{3.1}$$

$$X \equiv r_2 \bmod m_2 \tag{3.2}$$

$$X \equiv r_3 \bmod m_3 \tag{3.3}$$

$$X \equiv r_4 \bmod m_4 \tag{3.4}$$

Equation (1) can be written in an equation form as:

$$X = m_1 k + r_1 \tag{3.5}$$

Equation (5) must satisfy Equation (2) such that

$$m_1 k + r_1 \equiv r_2 \bmod m_2$$

$$m_1 k \equiv (r_2 - r_1) \bmod m_2$$

$$k \equiv (r_2 - r_1)\cdot m_1^{-1} \bmod m_2$$

$$k = m_2 t + \left|(r_2 - r_1)\cdot m_1^{-1}\right|_{m_2}.$$

Putting k into Equation (5), we have

$$X = m_1 m_2 t + \left(m_1\left|(r_2 - r_1) m_1^{-1}\right|_{m_2} + r_1\right) \tag{3.6}$$

Equation (6) must satisfy Equation (3) such that

$$m_1 m_2 t + \left(m_1\left|(r_2 - r_1) m_1^{-1}\right|_{m_2} + r_1\right) \equiv r_3 \bmod m_3$$

$$m_1 m_2 t \equiv r_3 - \left(m_1\left|(r_2 - r_1) m_1^{-1}\right|_{m_2} + r_1\right) \bmod m_3$$

$$t \equiv \left(r_3 - m_1\left|(r_2 - r_1) m_1^{-1}\right|_{m_2} + r_1\right)\cdot(m_1 m_2)^{-1} \bmod m_3$$

$$t = m_3 p + \left|\left(r_3 - m_1\left|(r_2 - r_1) m_1^{-1}\right|_{m_2} + r_1\right)\cdot(m_1 m_2)^{-1}\right|_{m_3}$$

Putting t into Equation (5), we have

$$X = m_1 m_2\left(m_3 p + \left|\left(r_3 - m_1\left|(r_2 - r_1) m_1^{-1}\right|_{m_2} + r_1\right)\cdot(m_1 m_2)^{-1}\right|_{m_3}\right) + \left(m_1\left|(r_2 - r_1) m_1^{-1}\right|_{m_2} + r_1\right)$$

$$X = m_1 m_2 m_3 p + m_1 m_2\left(\left|\left(r_3 - m_1\left|(r_2 - r_1) m_1^{-1}\right|_{m_2} + r_1\right)\cdot(m_1 m_2)^{-1}\right|_{m_3}\right) + \left(m_1\left|(r_2 - r_1) m_1^{-1}\right|_{m_2} + r_1\right) \tag{3.7}$$

Now Equation (7) is the general form of decimal equivalent that satisfies Equation (4) such that:

$$\left|m_1m_2m_3p+m_1m_2\left(\left|\left(r_3-m_1\left|\left(r_2-r_1\right)m_1^{-1}\right|_{m_2}+r_1\right)\cdot\left(m_1m_2\right)^{-1}\right|_{m_3}\right)\right.$$

$$\left.+\left(m_1\left|\left(r_2-r_1\right)m_1^{-1}\right|_{m_2}+r_1\right)\right|_{m_4}=r_4$$

where $p=\left[0,r_4\right]$.

Generally, the decimal equivalent for any four moduli set can be computed as follows:

X =

$$\left|m_1m_2m_3\mathrm{p}\ +\ m_1m_2\ (|(\ (r_3-s).(m_1m_2)^{-1}\ )|_{m_3})+\ s\right|_{m_4}\ =\ r_4$$

Where,

$$\mathrm{s}=\ \left(m_1|(r_2-\ r_1)m_1^{-1}|_{m_2}+\ r_1\right)$$

p = [0,r_4] that is, any value from the range that satisfies X after computing the value of *s*.

Therefore the value that satisfies X at a chosen number from p, is its decimal equivalent.

3.2. Algorithm for Five Moduli Set

Following the algorithm for four moduli set above,

The general decimal equivalent for any five moduli set $m=\{m_1,m_2,m_3,m_4,m_5\}$, with a given residues $r=(r_1,r_2,r_3,r_4,r_5)$ can be computed as follows:

X =

$$\left|m_1m_2m_3m_4\mathrm{p}\ +m_1m_2m_3(|(r_4-q).(m_1m_2m_3)^{-1}|_{m_4})+\ q\right|_{m_5}\ =\ r_5$$

Where,

$$\mathrm{q}=\ m_1m_2\ (|(\ (r_3-s).(m_1m_2)^{-1}\ )|_{m_3})+\ (s)$$

$$\mathrm{s}=\ \left(m_1|(r_2-\ r_1)m_1^{-1}|_{m_2}+\ r_1\right)$$

p = [0,r_5] that is, any value from the range that satisfies X after computing the values of *q* and *s*.

Therefore the value that satisfies X at a chosen number from p, is its decimal equivalent.

***Special case for four moduli sets with two or more of its residues being the same*:** Given a four moduli set $m=\left\{m_1,m_2,m_3,m_4\right\}$ and residues $r=\left(r_1,r_2,r_3,r_4\right)$, if any two of these residues are the same then the value of *s* in the formula for four moduli set can be directly set to the value of the residues that appeared twice without any computation of *s*.

Generally, the decimal equivalent for any four moduli set having two of its residues being the equal can be computed as follows:

X =

$$\left|m_1m_2m_3\mathrm{p}\ +\ m_1m_2\ (|(\ (r_3-s).(m_1m_2)^{-1}\ )|_{m_3})+\ s\right|_{m_4}\ =\ r_4$$

Where,

s, now is the value that appear twice in the residues.

p = [0,m_4-1] that is, any value from the range that satisfies X.

Therefore the value that satisfies X at a chosen number from p, is its decimal equivalent.

NOTE: $m_1\ \rightarrow\ r_1$ and $m_2\ \rightarrow\ r_2$ should be set for the two chosen residues that are equal together with their corresponding moduli.

With the advancement of four moduli set having two or more of its residues being the same, the number of multiplicative inverses and the number of arithmetic operations are further reduced.

NOTE: It should be noted that, the r_{i-1} which gives the valid range for *p* ($p=\left[0,r_{i-1}\right]$) can be chosen base on the least value of *m*m_i's in a given moduli set corresponding to its residue to limit the range of *p*.

4. Evaluation

Numerical evaluations of the algorithms are presented to ascertain the correctness of the algothms as follows:

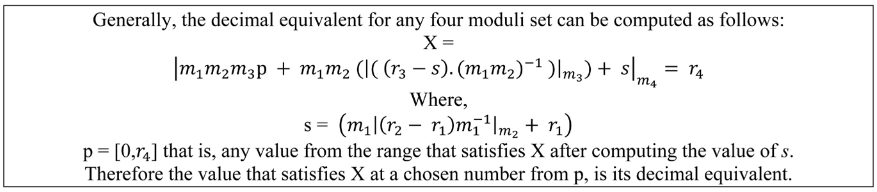
Consider the four-moduli set as $\{2^n+1, 2^n, 2^n-1, 2^{n-1}-1\}$.

As $n = 4$, we have $M = \{17, 16, 15, 7\}$ and residues $r = (7, 11, 0, 5)$.

As $n = 2$, we have $M = \{5, 4, 3, 1\}$ and residues $r = (2, 1, 0, 0)$.

Solution:

With four moduli set, we have:



Generally, the decimal equivalent for any four moduli set can be computed as follows:

X =

$$\left|m_1 m_2 m_3 \mathrm{p} + m_1 m_2 \left(|\left((r_3 - s).(m_1 m_2)^{-1}\right)|_{m_3}\right) + s\right|_{m_4} = r_4$$

Where,

$$\mathrm{s} = \left(m_1 |(r_2 - r_1) m_1^{-1}|_{m_2} + r_1\right)$$

p = [0,r_4] that is, any value from the range that satisfies X after computing the value of s.

Therefore the value that satisfies X at a chosen number from p, is its decimal equivalent.

From example 1,

$$m_1 = 17, m_2 = 16, m_3 = 15, m_4 = 7 r_1 = 7, r_2 = 11, r_3 = 0, r_4 = 5$$

$$X = \left|17 \times 16 \times 15 p + 17 \times 16 \left(\left|(0 - s) \times (17 \times 16)^{-1}\right|_{15}\right) + s\right|_7 = 5$$

where,

$$s = 17\left|(11 - 7) \times 17^{-1}\right|_{16} + 7$$

$$s = 17|4 \times 1|_{16} + 7$$

$$s = 75$$

$$X = \left|4080 p + 272\left(\left|(0 - 75) \times 272^{-1}\right|_{15}\right) + 75\right|_7 = 5$$

$$X = \left|4080 p + 272\left(\left|(-75) \times 8\right|_{15}\right) + 211\right|_7 = 5$$

$$X = |4080 p + 272 \times 0 + 75|_7 = 5$$

$$|4080 p + 75|_7 = 5$$

$$p = [0, 1, 2, 3, 4, 5].$$

When $p = 0$,

$|4080 \times 0 + 75|_7 = 5$ is satisfied. Since $|75|_7 = 5$ satisfied X at the value of 75 ($4080 \times 0 + 75$), we stop and take that value as its decimal equivalent.

Therefore the decimal equivalent for $n = 2$ is 75.

From example 2,

$$m_1 = 5, m_2 = 4, m_3 = 3, m_4 = 1, r_1 = 2, r_2 = 1, r_3 = 0, r_4 = 0$$

$$X = \left|5 \times 4 \times 3 p + 5 \times 4\left(\left|(0 - s) \times (5 \times 4)^{-1}\right|_3\right) + s\right|_1 = 0$$

where,

$$s = 5\left|(1 - 2) \times 5^{-1}\right|_4 + 2$$

$$s = 5\left|(-1)\times 1\right|_4 + 2$$

$$s = 5\times 3 + 2$$

$$s = 17$$

$$X = \left|60p + 20\left(\left|(0-17)\times 20^{-1}\right|_3\right) + 17\right|_1 = 0$$

$$X = \left|60p + 20\left(\left|(-17)\times 2\right|_3\right) + 17\right|_1 = 0$$

$$X = \left|60p + 20\times 2 + 17\right|_1 = 0$$

$$\left|60p + 57\right|_1 = 0$$

$$p = [0].$$

When $p = 0$,

$\left|60\times 0 + 57\right|_1 = 0$ is satisfied. Since $\left|57\right|_1 = 0$ satisfied X at the value of 57 ($60\times 0 + 57$), we stop and take that value as its decimal equivalent.

Therefore the decimal equivalent for *n* = 2 is 57.

Consider the five-moduli set as $\left\{2^{n-1}-1, 2^n+1, 2^n, 2^n-1, 2^{n+1}-1\right\}$.

As $n = 4$ we have $m = \{7, 17, 16, 15, 31\}$ and residues $r = (4, 14, 4, 10, 5)$.

As $n = 5$, we have $m = \{31, 65, 64, 63, 127\}$ and residues $r = (6, 6, 37, 5, 116)$.

Solution:

With five moduli set, we have:

The general decimal equivalent for any five moduli se t$m = \{m_1, m_2, m_3, m_4, m_5\}, with\ a$ given residues can be computed as follows:

X =

$$\left|m_1m_2m_3m_4\text{p} + m_1m_2m_3(|(r_4 - q).(m_1m_2m_3)^{-1}|_{m_4}) + q\right|_{m_5} = r_5$$

Where,

$$\text{q} = m_1m_2\,(|(\,(r_3 - s).(m_1m_2)^{-1}\,)|_{m_3}) + (s)$$

$$\text{s} = \left(m_1|(r_2 - r_1)m_1^{-1}|_{m_2} + r_1\right)$$

p = [0,r_5] that is, any value from the range that satisfies X after computing the values of q and s. Therefore the value that satisfies X at a chosen number from p, is its decimal equivalent.

We earlier stated that to limit the range of *p*, the smallest m_i can be chosen instead of using m_{i-1} for the range of *p*. Example 3 and 4 will be used for instance.

From example 3, the smallest modulus is 7 corresponding to residue 4 will be used for the range of *p* instead of the last modulus which is 31. For this instance,

$$m_1 = 17, m_2 = 16, m_3 = 15, m_4 = 31, m_5 = 7, r_1 = 14, r_2 = 4, r_3 = 10, r_4 = 5, r_5 = 4$$

$$X = \left|17\times 16\times 15\times 31p + 17\times 16\times 15\left(\left|(5-q)\times(17\times 16\times 15)^{-1}\right|_{31}\right) + q\right|_7 = 4$$

where,

$$s = 17\left|(4-14)\times 17^{-1}\right|_{16} + 14$$

$$s = 17\left|(-10)\times 1\right|_{16} + 14$$

$$s = 17\times 6 + 14$$

$$s = 116$$

$$q=17\times16\left(\left|(10-116)\times(17\times16)^{-1}\right|_{15}\right)+116$$

$$q=272\left(\left|(-106)\times8\right|_{15}\right)+116$$

$$q=272\times7+109$$

$$q=2020$$

$$X=\left|126480p+4080\left(\left|(5-2020)\times4080^{-1}\right|_{31}\right)+2020\right|_{7}=4$$

$$\left|126480p+4080\left(\left|(-2015)\times18\right|_{31}\right)+2020\right|_{7}=4$$

$$\left|126480p+4080\left(\left|-36270\right|_{31}\right)+2020\right|_{7}=4$$

$$\left|126480p+4080\times0+2020\right|_{7}=4$$

$$\left|126480p+2020\right|_{7}=4$$

$$p=[0,1,2,3,4].$$

When $p=0$,

$\left|126480\times0+2020\right|_{7}=4$ is satisfied. Since $\left|2020\right|_{7}=4$ satisfied X at the value of 2020 ($126480\times0+2020$), we stop and take that value as its decimal equivalent.

Therefore the decimal equivalent as *n* = 4 is 2020.

From example 4, the smallest modulus is 31 corresponding to residue 6 will be used for the range of p instead of the last modulus which is 127. For this instance, $m_1=65, m_2=64, m_3=63, m_4=127, m_5=31, r_1=6, r_2=37, r_3=5, r_4=116$, $r_5=6$

$$X=\left|65\times64\times63\times127p+65\times64\times63\left(\left|(116-q)\times(65\times64\times63)^{-1}\right|_{127}\right)+q\right|_{31}=6,$$

where,

$$s=65\left|(37-6)\times65^{-1}\right|_{64}+6$$

$$s=65\left|31\times1\right|_{64}+6$$

$$s=65\times31+6$$

$$s=2021$$

$$q=65\times64\left(\left|(5-2021)\times(65\times64)^{-1}\right|_{63}\right)+2021$$

$$q=4160\left(\left|(-2016)\times32\right|_{63}\right)+2021$$

$$q=4160\times0+2021$$

$$q=2021$$

$$X=\left|33284160p+262080\left(\left|(116-2021)\times262080^{-1}\right|_{127}\right)+2021\right|_{31}=6$$

$$\left|33284160p+262080\left(\left|(-1905)\times82\right|_{127}\right)+2021\right|_{31}=6$$

$$\left|33284160p + 262080\left(\left|-156210\right|_{127}\right) + 2021\right|_{31} = 6$$

$$\left|33284160p + 262080 \times 0 + 2021\right|_{31} = 6$$

$$\left|33284160p + 2021\right|_{31} = 6$$

$$p = [0,1,2,\cdots,6],$$

when $p = 0$

$\left|33284160 \times 0 + 2021\right|_{31} = 6$ is satisfied. Since $\left|2021\right|_7 = 4$ satisfied X at the value of 2021 ($33284160 \times 0 + 2021$) we stop and take that value as its decimal equivalent.

Therefore the decimal equivalent as $n = 5$ is 2021.

***Numerical illustration for special case of four moduli sets with two or more of its residues being the same*:**

As n = 2 in the four-moduli set, $m_1 = 5, m_2 = 4, m_3 = 3, m_4 = 1, r_1 = 2, r_2 = 1$, $r_3 = 0, r_4 = 0$. Two of its residues are the same. That is $r_3 = 0$ and $r_4 = 0$. Hence the value of s in the formula for four moduli set must be set to the value of r_3 or r_4 which is zero (0). Moreover, $r_3 = 0 \to m_3 = 3$ and $r_4 = 0 \to m_4 = 1$ should be set as $r_1 = 0 \to m_1 = 3$ and $r_2 = 0 \to m_2 = 1$. Therefore $m_1 = 3, m_2 = 1$, $m_3 = 5, m_4 = 4, r_1 = 0, r_2 = 0, r_3 = 2, r_4 = 1$. Hence,

$$X = \left|m_1 m_2 m_3 p + m_1 m_2\left(\left|\left(r_3 - 0\right) \cdot \left(m_1 m_2\right)^{-1}\right|_{m_3}\right) + 0\right|_{m_4} = r_4$$

$$X = \left|3 \times 1 \times 5p + 3 \times 1\left(\left|\left(2 - 0\right) \times \left(3 \times 1\right)^{-1}\right|_5\right) + 0\right|_4 = 1$$

$$X = \left|15p + 3\left(\left|2 \times 2\right|_5\right) + 0\right|_4 = 1$$

$$X = \left|15p + 3 \times 4 + 0\right|_4 = 1$$

$$X = \left|15p + 12\right|_4 = 1$$

$$p = [0,1,2,3].$$

When $p = 3$,

$\left|15 \times 3 + 12\right|_4 = 1$ is satisfied. Since $\left|57\right|_4 = 1$ satisfied X at the value of 57 ($15 \times 3 + 12$), we stop and take that value as its decimal equivalent.

Therefore the decimal equivalent is still 57 as required.

5. Conclusion

A new algorithm for reverse conversion for four-moduli set and a five-moduli set have been proposed. This will have a reverse conversion algorithm with high dynamic range, where more numbers can be uniquely and unambiguously represented. Numerical evaluation shows the correctness of the algorithm.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Tanaka, R.I. and Szabo, N.S. (1967) Residue Arithmetic and its Applications to Computer Technology. McGraw Hill, New York.

[2] Soderstrand, M.A., Jenkins, W.K, Jullien, G.A. and Taylor, F.J. (1986) Residue Number System Arithmetic: Modern Applications in Digital Signal Processing. IEEE Press, New York.

[3] 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.

[4] Daniel, A. and Abdul, M. (2021) New Algorithm for Reverse Conversion in Residue Number System. *Global Scientific Journals*, **9**, 661-665.

[5] Kalyanaraman, K. and Nicholas, C.H.V. (2013) Efficient Reverse Converters Designs for RNS based Digital Signal Processing Systems. *IEEE 2nd Global Conference on Consumer Electronics-GCCE*, 1-4 October 2013, Tokyo. https://doi.org/10.1109/GCCE.2013.6664781

[6] Azadeh, A.Z., Amir, S.M., *et.al.* (2014) Reverse Converter Design via Parallel-Prefix Adders: Novel Components, Methodology, and Implementations. *IEEE Transactions on Very Large Scale Integration* (*VLSI*) *Systems*, 2015, Kansas.

[7] Omondi, A. and Premkumar, B. (2007) Residue Number System: Theory and Implementation. Imperial College Press, London.

[8] Reddy, Y.A. 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.

[9] Duc-Minh, P., Premkumar, A.B. and Madhukumar, A.S. (2011) Error Detection and Correction in Communication Channels Using Inverse Gray RSNS Codes. *IEEE Transactions on Communications—TCOM*, **59**, 975-986. https://doi.org/10.1109/TCOMM.2011.022811.100092

[10] Omondi, A. and Premkumar, B. (2007) Residue Number Systems: Theory and Implementation. Imperial College Press, London. https://doi.org/10.1142/p523

[11] Chung-Kuan, C. (2006) Computer Arithmetic Algorithms and Hardware Design, Lecture notes. University of California, San Diego.