Science Frontiers
2023; 4(1): 8-16
http://www.sciencepublishinggroup.com/j/sf
doi: 10.11648/j.sf.20230401.12

Efficient Sign-Detection-Scheme Using Modular Computation Technique for the Moduli Set $\{2^n-1, 2^n, 2^n+1, 2^{(n+1)}-1, 2^{2n}-5\}$

Mohammed Ibrahim Daabo*, Valentine Aveyom

Department of Computer Science, C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana

Email address:
idaabo@cktutas.edu.gh (Mohammed Ibrahim Daabo)

*Corresponding author

To cite this article:
Mohammed Ibrahim Daabo, Valentine Aveyom. Efficient Sign-Detection-Scheme Using Modular Computation Technique for the Moduli Set $\{2^n-1, 2^n, 2^n+1, 2^{(n+1)}-1, 2^{2n}-5\}$. *Science Frontiers*. Vol. 4, No. 1, 2023, pp. 8-16. doi: 10.11648/j.sf.20230401.12

Received: MM DD, 2023; **Accepted**: MM DD, 2023; **Published**: MM DD, 2023

Abstract: In computer arithmetic, one of the most important things to consider in hardware design is the ability of the system to detect and display numbers with their signs. This when properly managed will reduce errors and ensure hardware reliability. But interestingly, detecting and knowing the sign of a residue number during arithmetic operation is very difficult. Magnitude Comparison, Scaling and Number conversions are some of the other difficult operations in Residue Number System (RNS). Unlike the weighted number system, it is even extremely difficult to determine the sign of a number in an RNS architecture thereby hampering the full implementation RNS in general purpose computing. In this paper, an efficient sign detection algorithm for detecting the sign of a number in an RNS architecture is presented. In formulating the algorithms, X maximum, (X_{max}) is computed from the Dynamic Range, $M=\prod^{k}_{i=1}(m_i)$. Modular Computation Technique is employed as a converter to compute X from the residues (r_1, r_2, r_3) with respect to a given moduli set, say $S= \{m_1, m_2 \ldots, m_n\}$. X is positive if $X-X_{max}<0$ otherwise X is negative and the actual value is this case is computed as X-M. The moduli set $\{2^n-1, 2^n, 2^n+1, 2^{(n+1)}-1, 2^{2n}-5\}$ is used for the system design implementation and for numerical illustrations. It is observed that the scheme effectively detects the sign of RNS numbers and theoretical analysis showed that simple hardware resources and low-power modular adders are used in the design. It is also observed that the scheme when implemented practically can help project RNS to be used in general purpose computing.

Keywords: RNS - Residue Number System, CRT - Chinese Remainder Theorem, MRC – Mixed Radix Conversion, Sign Detection

1. Introduction

In recent times, there has been much growing interest in the study of Residue Number System (RNS) in the field of parallel computing. This can be significantly traced to the great deal of computing that takes place in embedded processors such as those found in mobile devices and signal processing which normally require high speed computations with low-power consumption. The absence of carry-propagation in RNS results in parallel computing that guarantees fault-tolerance, high-speed and low-power arithmetic. Today, computer chips have become so dense that full testing is no longer possible and therefore has made fault-tolerance and the general area of computational integrity essential. In Digital Computer Arithmetic, this kind of number representation introduces parallelism Paharmi [6], which is very useful in applications like cryptography Antao and Sousa [3] and Digital Signal Processing Soderstand et al., [7]. Nonetheless, while operations such as addition, subtraction and multiplication may be carried out quickly and directly in parallel on the residues, additional arithmetic operations like reverse conversion, scaling, magnitude comparison and sign and overflow detection are challenging to implement in RNS Paharmi [6]. Traditionally, investigations to detect the sign and compare the magnitude of numbers largely relied on RNS reverse conversion techniques such as the Chinese Remainder Theorem (CRT) and the Mixed Radix Conversion (MRC) [9]. In this approach, numbers are translated from RNS into a positional number representation scheme where the comparison operation may be computed effectively.

1.1. Fundamentals of RNS

RNS is defined by a set *S,* of *N* integers that are pair-wise relatively prime. That is S= $\{m_1, m2, \ldots, m_n\}$ where the greatest common divisor of any pair (m_i, m_j) is 1. That is $\gcd(m_i, m_j) = 1$ for i,=1,..., N and $i \neq j$. In this case, every integer X in [0,−1], can be uniquely represented with an *N*-tuple where, $M = \prod_{i=1}^{k} m_i$ is the dynamic range. For any integer X, we have X→ $(x_{1,2},\ldots,x_N)$ and $x_i = |X|_{m_i}$= (X mod m_i); for i=1 to N. The set S and the number x_i are called the moduli set and residue of X modulo m_i respectively.

1.2. Application of RNS

One major breakthrough in the study of RNS is the discovery of its ability to carry out high speed computation as well as perform parallel arithmetic and processing. These attributes have led to its adoption in Digital Signal Processing applications such as Digital Filtering, Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), Digital Communication and Error Detection and Correction.

1.3. Challenges of RNS

Generally, architectures built on RNS have great advantage in terms of speed, fault-tolerance and power management. This therefore has made it very suitable to implement RNS-based processors in different applications. However, in spite of these great advantages, RNS processors do not find wide range usage in practical computing due to some limiting factors that make the process very difficult. Some of these factors that have resulted in open research are: Conversion, Magnitude Comparison, Sign Detection and Overflow Detection among others.

1.4. Domain of the Research

In this research, we propose an efficient sign-detection-scheme by employing the modular computation technique as a converter in the field of Residue Number System.

2. Literature Review

In RNS, it is difficult to identify a negative number and many researchers have looked into sign detecting algorithms as an alternative. In Ulman [10], a sign detection algorithm for a certain class of RNS was proposed which uses a sum of modulo 2 of digits in the related Mixed Radix System (MRS). Vu [11] proposed a sign detection strategy based on fractional representation for reducing the total modulo M in the conversion formula to a sum modulo 2. Xu et al., Al-Radadi [2], presented a sign identification technique based on the new Chinese Remainder Theorem (CRT) II. The modulo operations in the sign detection algorithm have a large size of modulo M. Again, another sign detection approach by Akkal and Siy [1] uses the nth Mixed Radix Digit (MRD) in Mixed-Radix Conversion (MRC). Hiasat [4], proposed a residue-based sign detection scheme using carry-save and carry- generation circuits on a four-moduli set to enhance the performance of an arithmetic unit. This reduced the time delay by a large margin and performance analysis with a similar sign detector put the proposed converter ahead in time, area and power consumption. VLSI tools were used to experiment the scheme to confirm its gain. In all, the research promised to be relevant in applications requiring high speed, such as communication systems. Akkal and Siy [1] presented a generic sign detection algorithm based on the Mixed Radix Conversion algorithm, MRC-II. The algorithm uses only one step comparison for sign detection. The conversion algorithm works by eliminating the need for table lookup normally used in MRC hardware implementation and hence does not need ROM as in the case of other algorithms. A fast RNS sign detection algorithm for the restricted moduli set $\{2^n\text{-}1,\ 2^n+1,\ 2^{2n}+1, 2^{n+k}\}$ has also been discussed by Xu et al., [12]. Their proposed algorithm allows for parallel implementation and consists exclusively of modulo 2n additions. The implementation of the algorithm has been done using one carry save adder, one comparator and one prefix adder. The experimental results showed that the proposed circuit unit had gained in area, delay and power by 63.8%, 44.9%, and 67.6% respectively when compared with a unit based on one of the best sign detection algorithms. In a paper authored by Antao and Sousa [3], a new approach was proposed for sign detection and number comparison using a revised version of the Mixed Radix Conversion (MRC) for an augmented 3-moduli sets $\{2^n+1, 2^n-1, 2^{n+k}\}$. Almost all their computations were directly performed on the moduli channels, allowing for easy use in any RNS processor. The paper further presented an efficient, unified and very large-scale integration architecture based on the proposed scheme and this was implemented using 65 nm CMOS technologies which showed that the proposed architecture was more efficient than some related state of the art architecture. Sousa and Martins [8] designed an efficient RNS comparators for large dynamic range and implemented it on the $\{2^n + 1, 2^n - 1, 2^{n+x}\}$ moduli set. The design technique followed a two-step approach. Firstly, the MRC was used to optimize the sign identification, by focusing on the extraction of the sign bit and on the parallelism of the hardware structure. The sign identification module was then used to implement the number comparison operation. The proposed comparator demonstrated to be a very attractive RNS-Based hardware architecture for signal processing applications. Hiasat [5] introduced a new architecture for sign detection for the moduli set $\{2^{n+p}, 2^n - 1, 2^n + 1,\}$, where n and p are positive integers such that p < n. The proposed decoder was seen to be flexible and efficient since it could easily deal with large dynamic range starting from 3n bits up to 3n + p bits dynamic range. For cases where p > 1, increasing the dynamic range by 1 bit (i.e., doubling the dynamic range) requires only one half-adder and no additional delay. When compared with the most competitive published work, the new sign detector exhibited a better performance in terms of area, delay and power that ranges from 4.7 percent up to 44.8 percent. Younes and Steffan [13] presented two designs for overflow and sign detection and correction in unsigned and signed RNS based on the moduli set $\{2n - 1, 2n, 2n + 1\}$. This set has an even dynamic range. Moreover, these designs can be considered as universal, since they can be used with any system that has an even dynamic range by applying a small

modification on the evaluation unit. Both designs are faster and require less hardware components than those based on comparators.

3. Methodology

3.1. Proposed Algorithm

In this section, we propose an efficient algorithm to determine the sign of an RNS number by dividing the procedure into two parts. In part one, we compute X_{max} and then in part two, we compute the decimal equivalent, X of a given residue number, say $X=(r_1, r_2, \dots, r_n)$ with respect to the moduli set $S = \{m_1, m_2, \dots, m_n\}$ using the modular computation technique as a converter. We then perform the necessary comparisons in order to determine the sign of the RNS number. The sign detection process is illustrated as follows:

$$\text{Compute } X_{max} = \frac{M}{2} - 1$$

where $M = \prod_{i=1}^{k} m_i$

Compute X using the modular computation method as a reverse converter for a given residue number, $X=(r_1, r_2, \dots, r_n)$ with respect to the moduli set $S = \{m_1, m_2, \dots, m_n\}$

Decision Whether Sign is Positive OR Negative

If the decimal number, X computed is less than X_{max} then it is positive.

However, If the decimal number, X computed is greater than X_{max} then, we determine β=X-M. In this case β will be negative.

3.2. Modular Computation Technique

The Modular Computation Method is a fast but low power dispense Residue-to-binary conversion technique that employs modular computations in the reverse conversion process. Given a relatively prime moduli set $\{m_1, m_2, \dots, m_n\}$ and its corresponding residues $(r_1, r_2, \dots, r_n)$, we define the first jump j_1, which is equal to the first residue r_1, and l_1 as its first location given by:

1st Jump: $J_1 = r_1$

1st location $L_1 = |X - J_1| = \begin{bmatrix} |r_1 - r_1|_{m_1} = 0 \\ |r_2 - r_1|_{m_2} = {r^i}_2 \\ \cdot \\ \cdot \\ \cdot \\ |r_n - r_1|_{m_n} = {r^i}_n \end{bmatrix}$

2nd Jump: $J_2 = m_1K_2$ and $\left|{r^i}_2 - J_2\right|_{m_2} = 0$ $\left|{r^i}_2 - m_1K_2\right|_{m_2} = 0$ $\Rightarrow K_2 = \left|\frac{r_2^1}{m_1}\right|_{m_2}$

2nd location $L_2 = |X - J_1| = \begin{bmatrix} |r_1 - r_1|_{m_1} = 0 \\ |r_2 - r_1|_{m_2} = 0 \\ \cdot \\ \cdot \\ \cdot \\ |r_n - r_1|_{m_n} = {r^i}_n \end{bmatrix}$

We continue finding the next jump and location and stop when $n^{th} location = 0$

Such that n^{th} location $L_n = |X - J_n| = \begin{bmatrix} |r_1 - r_1|_{m_1} = 0 \\ |r_2 - r_1|_{m_2} = 0 \\ \cdot \\ \cdot \\ \cdot \\ |r_n - r_1|_{m_n} = 0 \end{bmatrix}$

Finally, the corresponding decimal number X is the result of summing of the $J_i's$, thus $X = J_1 + J_2 + \cdots + J_n$

3.3. Proposed Converter

The proposed algorithm using cyclic jump method is presented in details below:

Given a 5-moduli set $S = \{m_1, m_2, m_3, m_4, m_5\}$ such that $\{r_1, r_2, r_3, r_4, r_5\}$ are the residues of the decimal number X, then by the cyclic jump approach we have;

1st Jump: $J_1 = r_1$

1st location $L_1 = |X - J_1| = \begin{bmatrix} |r_1 - r_1|_{m_1} = 0 \\ |r_2 - r_1|_{m_2} = {r^i}_2 \\ |r_3 - r_1|_{m_3} = {r^i}_3 \\ |r_4 - r_1|_{m_4} = {r^i}_4 \\ |r_5 - r_1|_{m_5} = {r^i}_5 \end{bmatrix}$

2nd Jump: $J_2 = m_1K_2$ and $\left|r^i{}_2 - J_2\right|_{m_2} = 0$ $\left|r^i{}_2 - m_1K_2\right|_{m_2} = 0 \Rightarrow K_2 = \left|\frac{r_2^i}{m_1}\right|_{m_2}$

2nSd location $L_2 = |X - J_2| = \begin{bmatrix} |0 - (m_1K_2)|_{m_1} = 0 \\ \left|r^i{}_2 - (m_1K_2)\right|_{m_2} = 0 \\ \left|r^i{}_3 - (m_1K_2)\right|_{m_3} = r^{ii}{}_3 \\ \left|r^i{}_4 - (m_1K_2)\right|_{m_4} = r^{ii}{}_4 \\ \left|r^i{}_5 - (m_1K_2)\right|_{m_5} = r^{ii}{}_5 \end{bmatrix}$

3rd Jump: $J_3 = m_1m_2K_3$ and $\left|r^{ii}{}_3 - J_3\right|_{m_3} = 0$ $\left|r^{ii}{}_3 - m_1m_2K_3\right|_{m_3} = 0 \Rightarrow K_3 = \left|\frac{r^{ii}{}_3}{m_1m_2}\right|_{m_3}$

3rd Location $L_3 = |X - J_3| = \begin{bmatrix} |0 - (m_1m_2K_3)|_{m_1} = 0 \\ |0 - (m_1m_2K_3)|_{m_2} = 0 \\ \left|r^{ii}{}_3 - (m_1m_2K_3)\right|_{m_3} = 0 \\ \left|r^{ii}{}_4 - (m_1m_2K_3)\right|_{m_4} = r^{iii}{}_4 \\ \left|r^{ii}{}_5 - (m_1m_2K_3)\right|_{m_5} = r^{iii}{}_5 \end{bmatrix}$

4th Jump: $J_3 = m_1m_2m_3K_4$ and $\left|r^{iii}{}_4 - J_4\right|_{m_4} = 0 \Rightarrow K_4 = \left|\frac{r^{iii}{}_4}{m_1m_2m_3}\right|_{m_4}$

4th Location $L_4 = |X - J_4| = \begin{bmatrix} |0 - (m_1m_2m_3K_4)|_{m_1} = 0 \\ |0 - (m_1m_2m_3K_4)|_{m_2} = 0 \\ |0 - (m_1m_2m_3K_4)|_{m_3} = 0 \\ \left|r^{iii}{}_4 - (m_1m_2m_3K_4)\right|_{m_4} = 0 \\ \left|r^{iii}{}_5 - (m_1m_2m_3K_4)\right|_{m_5} = r^{iv}{}_5 \end{bmatrix}$

5th Jump: $J_5 = m_1m_2m_3m_4K_5$ and $\left|r^{iv}{}_5 - J_5\right|_{m_5} = 0 \Rightarrow K_5 = \left|\frac{r^{iv}{}_5}{m_1m_2m_3m_4}\right|_{m_5}$

5th Location $L_5 = |X - J_5| = \begin{bmatrix} |0 - (m_1m_2m_3m_4K_5)|_{m_1} = 0 \\ |0 - (m_1m_2m_3m_4K_5)|_{m_2} = 0 \\ |0 - (m_1m_2m_3m_4K_5)|_{m_3} = 0 \\ |0 - (m_1m_2m_3m_4K_5)|_{m_4} = 0 \\ |0 - (m_1m_2m_3m_4K_5)|_{m_5} = 0 \end{bmatrix}$

At this point $(r^v{}_1, r^v{}_2, r^v{}_3, r^v{}_4, r^v{}_5) = (0,0,0,0,0)$

Finally, the corresponding decimal number X is the result of summing of the $J_i's$, thus $X = J_1 + J_2 + J_3 + J_4 + J_5$

4. Results and Discussions

4.1. Implementation of the Algorithm on the Moduli Set $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1, 2^{2n} - 5\}$

Given a 5-moduli set $S = \{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1, 2^{2n} - 5\}$ where $m_1 = 2^n - 1, m_2 = 2^n, m_3 = 2^n + 1, m_4 = 2^{n+1} - 1, m_5 = 2^{2n} - 5$ such that (r_1, r_2, r_3, r_4, r_5) are the residues of the decimal number X.

1st Jump: $J_1 = r_1$

1st location $L_1 = |X - J_1| = \begin{bmatrix} |r_1 - r_1|_{2^n-1} = 0 \\ |r_2 - r_1|_{2^n} = r^i{}_2 \\ |r_3 - r_1|_{2^n+1} = r^i{}_3 \\ |r_4 - r_1|_{2^{n+1}-1} = r^i{}_4 \\ |r_5 - r_1|_{2^{2n}-5} = r^i{}_5 \end{bmatrix}$

2nd Jump: $J_2 = (2^n - 1)K_2$ and $\left|r^i{}_2 - J_2\right|_{2^n+1} = 0$ $\left|r^i{}_2 - (2^n - 1)K_2\right|_{2^n-1} = 0 \Rightarrow K_2 = \left|\frac{r_2^1}{2^n-1}\right|_{2^n-1}$

2nd location $L_2 = |X - J_2| = \begin{bmatrix} |0 - ((2^n - 1)K_2)|_{2^n-1} = 0 \\ \left|r^i{}_2 - ((2^n - 1)K_2)\right|_{2^n} = 0 \\ \left|r^i{}_3 - ((2^n - 1)K_2)\right|_{2^n+1} = r^{ii}{}_3 \\ \left|r^i{}_4 - ((2^n - 1)K_2)\right|_{2^{n+1}-1} = r^{ii}{}_4 \\ \left|r^i{}_5 - ((2^n - 1)K_2)\right|_{2^{2n}-5} = r^{ii}{}_5 \end{bmatrix}$

3rd Jump: $J_3 = (2^n-1)(2^n)K_3$ and $\left|r^{ii}{}_3 - J_3\right|_{2^n+1} = 0$ $\left|r^{ii}{}_3 - (2^{2n+1})(2^{2n}+1)K_3\right|_{2^n+1} = 0 \Rightarrow K_3 = \left|\frac{r^{ii}{}_3}{(2^n-1)(2^n)}\right|_{2^n+1}$

3rd Location $L_3 = |X - J_3| = \begin{bmatrix} \left|0 - ((2^n-1)(2^n)K_3)\right|_{2^n-1} = 0 \\ \left|0 - ((2^n-1)(2^n)K_3)\right|_{2^n} = 0 \\ \left|r^{ii}{}_3 - ((2^n-1)(2^n)K_3)\right|_{2^n+1} = 0 \\ \left|r^{ii}{}_4 - ((2^n-1)(2^n)K_3)\right|_{2^{n+1}-1} = r^{iii}{}_4 \\ \left|r^{ii}{}_5 - ((2^n-1)(2^n)K_3)\right|_{2^{2n}-5} = r^{iii}{}_5 \end{bmatrix}$

4th Jump: $J_3 = (2^n-1)(2^n)(2^n+1)K_4$ and $\left|r^{iii}{}_4 - J_4\right|_{2^{2n}-5} = 0 \Rightarrow K_4 = \left|\frac{r^{iii}{}_4}{(2^n-1)(2^n)(2^n+1)}\right|_{2^{2n}-5}$

4th Location $L_4 = |X - J_4| = \begin{bmatrix} \left|0 - ((2^n-1)(2^n)(2^n+1)K_4)\right|_{2^n-1} = 0 \\ \left|0 - ((2^n-1)(2^n)(2^n+1)K_4)\right|_{2^n} = 0 \\ \left|0 - ((2^n-1)(2^n)(2^n+1)K_4)\right|_{2^n+1} = 0 \\ \left|r^{iii}{}_4 - ((2^n-1)(2^n)(2^n+1)K_4)\right|_{2^{n+1}-1} = 0 \\ \left|r^{iii}{}_5 - ((2^n-1)(2^n)(2^n+1)K_4)\right|_{2^{2n}-5} = r^{iv}{}_5 \end{bmatrix}$

5th Jump: $J_5 = (2^{n+1})(2^n-1)(2^n+1)(2^{n+1}-1)K_5$ and $\left|r^{iv}{}_5 - J_5\right|_{2^{2n}-1} = 0$

$$\Rightarrow K_5 = \left|\frac{r^{iv}{}_5}{(2^n-1)(2^n-1)(2^n+1)(2^{n+1}-1)}\right|_{2^{2n}-1}$$

5th Location $L_5 = |X - J_5| = \begin{bmatrix} \left|0 - ((2^n-1)(2^n)(2^n+1)(2^{n+1}-1)K_5)\right|_{2^n-1} = 0 \\ \left|0 - ((2^n-1)(2^n)(2^n+1)(2^{n+1}-1)K_5)\right|_{2^n} = 0 \\ \left|0 - ((2^n-1)(2^n)(2^n+1)(2^{n+1}-1)K_5)\right|_{2^n+1} = 0 \\ \left|0 - ((2^n-1)(2^n)(2^n+1)(2^{n+1}-1)K_5)\right|_{2^{n+1}-1} = 0 \\ \left|0 - ((2^n-1)(2^n)(2^n+1)(2^{n+1}-1)K_5)\right|_{2^{2n}-5} = 0 \end{bmatrix}$

Since $(r^v{}_1, r^v{}_2, r^v{}_3, r^v{}_4, r^v{}_5) = (0,0,0,0,0)$

Therefore, the corresponding decimal number $X = J_1 + J_2 + J_3 + J_4 + J_5$

4.2. Numerical Illustration

Negative Sign Detection using the Algorithm

(A) For the given moduli set $S = \{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1, 2^{2n} - 5\}$ and taking $n = 2$ let the residue for the decimal number X, be $(1,1,4,3,0)$. Thus $m_1 = 3, m_2 = 4, m_3 = 5, m_4 = 7, m_5 = 11, r_1 = 1, r_2 = 1, r_3 = 4, r_4 = 3, r_5 = 0$.

i. The first jump J_1 is defined by the number which normally corresponds to the first residue in X. Thus $J_1 = 1$.

The first location L_1 is defined by $(X - J_1)$ $L_1 = (X-1) = \begin{bmatrix} |1-1|_3 = 0 \\ |1-1|_4 = 0 \\ |4-1|_5 = 3 \\ |3-1|_7 = 2 \\ |0-1|_{11} = 10 \end{bmatrix}$

ii. Second jump is defined by the number J_2, such that:

$J_2 = 3K_2$ and $|0 - 8K_2|_3 \Rightarrow K_2 = \left|\frac{0}{3}\right|_4 = 0$ Thus $J_2 = 0$

The second location L_2 is defined by $(X - J_2)L_2 = (X-0) = \begin{bmatrix} |0-0|_3 = 0 \\ |0-0|_4 = 0 \\ |3-0|_5 = 3 \\ |2-0|_7 = 2 \\ |10-0|_{11} = 10 \end{bmatrix}$

iii. Third jump is defined by the number $9J_3$, such that:

$J_3 = 3 * 4K_3$ and $|3 - 12K_3|_5 \Rightarrow K_3 = \left|\frac{3}{12}\right|_5 = 4$ Thus $J_3 = 12 * 4 = 48$

The third location L_3 is defined by $(X - J_3)$, $L_3 = (X - 48) = \begin{bmatrix} |0-48|_3 = 0 \\ |0-48|_4 = 0 \\ |3-48|_5 = 0 \\ |2-48|_7 = 3 \\ |10-48|_{11} = 6 \end{bmatrix}$

iv. Fourth jump is defined by the number J_4, such that:

$J_4 = 3 * 4 * 5K_4$ and $|3 - 60K_4|_7 \Rightarrow K_4 = \left|\frac{3}{60}\right|_7 = \left|\frac{1}{20}\right|_7 = \left|\frac{3}{4}\right|_7 = \left|\frac{5}{2}\right|_7 = 6$ Thus $J_4 = 60 * 6 = 360$

The fourth location L_4 is defined by $(X - J_4), L_4 = (X - 360) = \begin{bmatrix} |0-360|_3 = 0 \\ |0-360|_4 = 0 \\ |0-360|_5 = 0 \\ |3-360|_7 = 0 \\ |6-360|_{11} = 9 \end{bmatrix}$

Fifth jump is defined by the number J_5, such that:

$J_5 = 3 * 4 * 5 * 7K_5$ and $|9 - 420K_5|_{11} \Rightarrow K_5 = \left|\frac{9}{420}\right|_{11} = \left|\frac{42}{420}\right|_{11} = \left|\frac{9}{2}\right|_{11} = \left|\frac{20}{2}\right|_{11} = 10$ Thus $J_5 = 420 * 10 = 4200$

The fifth location L_5 is defined by $(X - J_5)$, $L_5 = (X - 120) = \begin{bmatrix} |0-4200|_3 = 0 \\ |0-4200|_4 = 0 \\ |0-4200|_5 = 0 \\ |0-4200|_7 = 0 \\ |9-4200|_{11} = 0 \end{bmatrix}$

Therefore, the corresponding equivalent decimal number X is: $J_1 + J_2 + J_3 + J_4 + J_5 = 1 + 0 + 48 + 360 + 4200 = 4609$

Thus $(1, 1, 4, 3, 0)_{RNS} = 4609_{decimal}$.

$$Dynamic\ range = M = \prod m_i = 3.4.5.7.11 = 4620$$

$$X_{max} = \frac{M}{2} - 1$$
$$= \frac{4620}{2} - 1$$
$$= 2310 - 1$$
$$= 2309$$

Comparing the X computed and the X_{max}, we have 4609 being greater than 2309 as can be seen in Table.1. We then subtract the dynamic range from the decimal number to get its equivalent sign number which is $4609 - 4620 = -11$. This indicates that, the number whose residues are $(1, 1, 4, 3, 0)$ is -11

Positive Sign Detection using the Algorithm

(B) For the given moduli set t $S = \{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1, 2^{2n} - 5\}$ and taking $n = 2$ let the residue for the decimal number X, be$(2, 3, 1, 4, 0)$. Thus $m_1 = 3, m_2 = 4, m_3 = 5, m_4 = 7, m_5 = 11, r_1 = 2, r_2 = 3, r_3 = 1, r_4 = 4, r_5 = 0$.

i) The first jump J_1 is defined by the number which normally corresponds to the first residue in X. Thus $J_1 = 2$.

The first location L_1 is defined by $(X - J_1)$, $L_1 = (X - 2) = \begin{bmatrix} |2-2|_3 = 0 \\ |3-2|_4 = 1 \\ |1-2|_5 = 4 \\ |4-2|_7 = 2 \\ |0-2|_{11} = 9 \end{bmatrix}$

ii) Second jump is defined by the number J_2, such that:

$J_2 = 3K_2$ and $|1 - 3K_2|_4 \Rightarrow K_2 = \left|\frac{1}{3}\right|_4 = \left|\frac{9}{3}\right|_4 = 3$, Thus $J_2 = 3 * 3 = 9$

The second location L_2 is defined by $(X - J_2)L_2 = (X - 9) = \begin{bmatrix} |0-9|_3 = 0 \\ |1-9|_4 = 0 \\ |4-9|_5 = 0 \\ |2-9|_7 = 0 \\ |9-9|_{11} = 0 \end{bmatrix}$

iii) Third jump is defined by the number J_3, such that:

$J_3 = 3 * 4K_3$ and $|0 - 12K_3|_5 \Rightarrow K_3 = \left|\frac{0}{12}\right|_5 = 0$ Thus $J_3 = 12 * 0 = 0$

The third location L_3 is defined by $(X - J_3)$ $L_3 = (X - 0) = \begin{bmatrix} |0-0|_3 = 0 \\ |0-0|_4 = 0 \\ |0-0|_5 = 0 \\ |0-0|_7 = 0 \\ |0-0|_{11} = 0 \end{bmatrix}$

iv) Fourth jump is defined by the number J_4, such that:

$J_4 = 3 * 4 * 5K_4$ and $|0 - 60K_4|_7 \Rightarrow K_4 = \left|\frac{0}{60}\right|_7 = 0$ Thus $J_4 = 60 * 0 = 0$

The fourth location L_4 is defined by $(X - J_4), L_4 = (X - 0) = \begin{bmatrix} |0-0|_3 = 0 \\ |0-0|_4 = 0 \\ |0-0|_5 = 0 \\ |0-0|_7 = 0 \\ |0-0|_{11} = 0 \end{bmatrix}$

v) Fifth jump is defined by the number J_5, such that:

$J_5 = 3 * 4 * 5 * 7K_5$ and $|0 - 420K_5|_{11} \Rightarrow K_5 = \left|\frac{0}{420}\right|_{11} = 0$ Thus $J_5 = 420 * 0 = 0$

The fifth location L_5 is defined by $(X - J_5), L_5 = (X - 0) = \begin{bmatrix} |0-0|_3 = 0 \\ |0-0|_4 = 0 \\ |0-0|_5 = 0 \\ |0-0|_7 = 0 \\ |0-0|_{11} = 0 \end{bmatrix}$

vi) Therefore, the corresponding equivalent decimal number X is: $J_1 + J_2 + J_3 + J_4 + J_5 = 9 + 2 + 0 + 0 + 0 = 11$

Thus $(2, 3, 1, 4, 0)_{RNS} = 11_{decimal}$.

Since 11 is less than 2309, it is a positive 11.

Comparing the X computed and the X_{max}, We have 11 being less than 2309 as seen in Table 1, we do not subtract the dynamic range from the decimal number to get its equivalent sign number, instead we take the results. This indicates that, the number whose residues are $(2, 3, 1, 4, 0)$ is 11.

Table 1. *Sign Detection on the Moduli Set {3, 4, 5, 7, 11}.*

X	m_i	$m_1 = 3$	$m_2 = 4$	$m_3 = 5$	$m_4 = 7$	$m_5 = 11$
$M = 4620$	**Integer X**	$\|X\|_{m1} = X_1$	$\|X\|_{m2} = X_2$	$\|X\|_{m3} = X_3$	$\|X\|_{m4} = X_4$	$\|X\|_{m5} = X_5$
0	0	0 j=0	0 j=0	0	0	0 j=0
1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	3	0	3	3	3	3
4	4	1	0	4	4	4
5	5	2	1	0	5	5
6	6	0	2	1	6	6
7	7	1	3	2	0	7
8	8	2	0	3	1	8
9	9	0	1	4	2	9
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
2304	2304	0	0	4	1	5
2305	2305	1	1	0	2	6
2306	2306	2	2	1	3	7
2307	2307	0	3	1	4	8
2308	2308	1	0	3	5	9
$X_{max} = 2309$	2309	2 j_{max}=769	1 j_{max}=577	4	6	10 j_{max}=209
2310	-2310	0 j= 770	2	0	0	0
2311	-2309	1	3	1	1	1
2312	-2308	2	0	2	2	2
2313	-2307	0	1	3	3	3
2314	-2306	1	2	4	4	4
2315	-2305	2	3	0	5	5
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
4617	-3	0	1	2	4	8
4618	-2	1	2	3	5	9
4619	-1	2 j=1539	3 j=2309	4	6	10 j=419

5. Hardware Implementation of the Proposed Scheme

In Figure 1, it can be seen that the proposed scheme is a simple architecture built mainly on Carry Propagation modular Adders (CPA-modm), two Carry Save Adders (2 CSA_S), two Accumulators ($2ACC_S$) and one Multiplier (1MUX). In determining K_n, the MUX and ACC are used whilst CPA-modms are used for all the modular computations in the conversion process. Also the CSA_s are used in processing the final X for the sign detection.

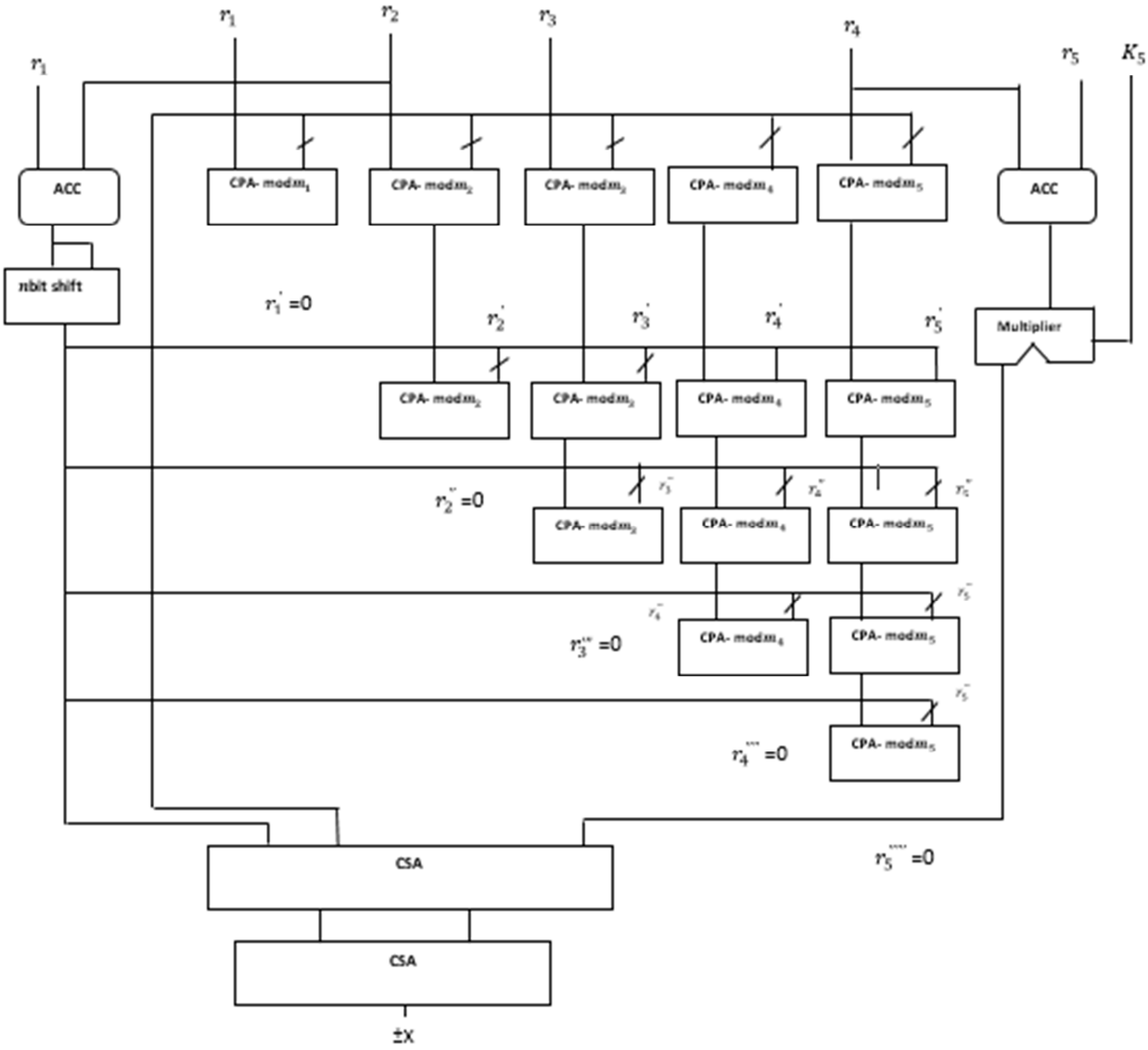

Figure 1. *Architecture for the Proposed Method.*

6. Performance Analysis

Table 2. *Comparing the Proposed Scheme with the CRT and MRC.*

Conversion Method	Use of Dynamic Range (M)	Sequential in Nature	Use of Modular Computation
CRT	YES	NO	LESS
MRC	NO	YES	LESS
Proposed Technique	NO	NO	DOMINATED

In Table 2, we compare the proposed scheme with the Chinese Remainder Theorem (CRT) and the Mixed Radix Conversion (MRC). It is observed that converters built around the CRT technology largely depend on the Dynamic Range (M) which is computationally intensive there by increasing the delay and power consumption. Similarly, converters built on the MRC are sequential in nature. Therefore, the computational processes are equally slow and

also inflict high power usage. In the proposed scheme, the conversion processes are dominated by modular computations which are traded for less delay, low power consumption and low cost (less resource requirements).

7. Conclusion

In this study, an efficient sign detection scheme in RNS has been proposed. The modular computation technique was used as a converter which resulted in high degree of parallelism and less computational intensity. Theoretical analysis showed that the scheme is advantaged with low power consumption, less delay and low cost of design. The scheme detects the sign of RSN numbers and can be implemented practically to help realize the dream of RNS being used in general purpose computing

References

[1] Akkal, M., & Siy, P. (2008). *Optimum RNS sign detection algorithm using MRC-II with special moduli set.* Journal of Systems Architecture, 54 (10), 937-944.

[2] Al-Radadi, E., & Siy, P. (2003). *RNS sign detector based on Chinese remainder theorem II (CRT II)*. Computers & Mathematics with Applications, 46 (10-11), 1559-1570. doi: 10.1016/S0898-1221(03)90147-7.

[3] Antão, S., & Sousa, L. (2013). *The CRNS framework and its application to programmable and reconfigurable cryptography.* ACM Transactions on Architecture and Code Optimization, 9 (4), 1-25. doi: 10.1145/2535918.

[4] Hiasat, A. (2018). *Sign detector for the extended four-moduli set* $\{2^n-1, 2^n+1, 2^{2n}+1, 2^{n+k}\}$. Computers & Digital Techniques, 32 (2), 69-73. https://doi.org/10.1049/iet-cdt.2017.0088.

[5] Hiasat, A. (2016). A Sign Detector for a Group of Three-Moduli Sets. IEEE Transactions on Computers, 65 (12), 3580–3590. https://doi.org/10.1109/TC.2016.2547381

[6] Parhami, B. (2010). *Computer Arithmetic: Algorithms and Hardware Designs (2nd ed.).* Oxford University Press.

[7] Soderstrand, M., Jenkins, W., Jullien, G., & Taylor, F. (Eds.). (1986). *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing.* IEEE Press. Piscataway, NJ.

[8] Sousa, L. & Martins, P. (2015). Sign Detection and Number Comparison on RNS 3-Moduli Sets. Springer Science+Business Media New York.

[9] Szabo, N., & Tanaka, R. (Eds.). (1967). *Residue arithmetic and its application to computer technology*. McGraw-Hill. New York.

[10] Ulman, Z. (1983). *Sign detection and implicit-explicit conversion of numbers in residue arithmetic*. IEEE Transactions on Computers, 32 (6), 590-594. doi: 10.1109/TC.1983.1676245.

[11] Vu, T. V. (1985). *Efficient implementations of the Chinese remainder theorem for sign detection and residue decoding*. IEEE Transactions on Computers, 34 (7), 646-651. doi: 10.1109/TC.1985.1676577.

[12] Xu, M., Bian, Z., & Yao, R. (2015). *Fast sign detection algorithm for the RNS moduli set* $\{2^{n+2}-1, 2^n-1, 2^n\}$. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 23 (2), 379-383.

[13] Younes, D., & Steffan, P. (2013). Universal approaches for overflow and sign detection in residue number system based on $\{2^n-1, 2^n, 2^n+1\}$. The Eighth International Conference on Systems (ICONS 2013).