

Residue Number system Sign Detector Design based on Dynamic Range Partitioning

A. Shahrbanooonezhad, A. S. Molahosseini, A. A. E. Zarandi, and M. Mohammadi

Abstract— The comparison of signed integers is needed in various applications such as signal processing, sorting and process control. However, sign detection is a complex operation in Residue Number System (RNS). The RNS magnitude comparison and sign detection algorithms can be categorized into two categories: Parity check-based algorithms and partial reverse conversion-based algorithms. This paper presents an effective method for RNS sign detection based on a new representation of negative numbers range in RNS. The proposed method is applied on a four-moduli set $\{2^{2n}, 2^n-1, 2^{n+1}, 2^{n+1}-1\}$ which resulted in an efficient sign detector with power-consumption and area saving than conventional sign detectors.

Index Term—partial reverse converter, RNS, sign detection, dynamic range partitioning

I. INTRODUCTION

Residue number system (RNS) is a non-weighted number system which can lead to parallel arithmetic [1]. The main advantage of this number system is that it offers the possibility of doing parallel and distributed calculations without dependence on carry bit propagation, which thus speeds up the calculation [2]-[3]. These advantages have led to various studies on this numerical system in the field of designing finite impulse response (FIR) filters as well as accelerators for deep learning [5], and other applications [8]. Beside the abovementioned advantages for residue number system, there are some limitations and problems for developing this system such as high complexity of sign detection, comparing the magnitude of numbers, overflow detection in calculations, and performing division [4]-[23].

Up to now, all of RNS sign detection algorithms have been based on dividing the dynamic range into two parts, and then comparing the operand with the half of dynamic range to determine each half it is placed to consequently determine the sign. [4]-[6].

Methods for sign detection and comparing the magnitude of numbers in residue number system fall into two general

categories:

1- Methods using the parity bit [24]

2- Partial reverse conversion-based methods [7]-[26]

There are various studies and algorithms for RNS sign detection in each of these categories. In this regard, [9] considered $Z=|X-Y|_M$ as $Z= (|2(x_1-y_1)-\dots-2(x_N-y_N)|)_{MN}$. Then based on the final parity bit and whether Z is even or not, the numbers X and Y can be compared.

In [10], an algorithm has been proposed to compare the unsigned numbers and signed ones based on extracting the sign of operands, and if signs are different then, the positive number will be greater. If operands have the same sign and are positive, algorithm [9] is used, and in the case where both are negative, their absolute values are calculated and then the algorithm mentioned in [9] will be used. It should be noted that in most of the algorithms which are based on the value of parity bit, an additional moduli 2 is used and if the examined number for moduli 2 is equal to zero, that number will be even.

The algorithm of [4] falls into the partial reverse conversion-based category since Chinese remainder theorem 2 (CRT-II) is used for sign detection and comparing the magnitude of numbers. However, it is limited to the common three moduli set $\{2^n-1, 2^n, 2^{n+1}\}$ which will have in turn a limited dynamic range. In [11] and [12], two algorithms have been presented for sign detection and magnitude comparison based on partial reverse conversion that uses CRT-II.

In [13], a new method based on an optimized version of MRC on the three-moduli set $\{2^{n-1}, 2^{n+x}, 2^{n+1}\}$ has been presented which can be used for sign detection, magnitude comparison, and overflow detection.

It should be noted that in some of these algorithms including [14], full reverse conversion has been used for converting a residue number into its binary equivalent as well as for comparing it to the half of dynamic range for sign detection. Therefore, [14] suffers from high energy consumption and delay due to the need for full reverse conversion

In [15] an algorithm based on mixed radix CRT is suggested for sign detection with assuming there based on an n -moduli set $\{x_1, \dots, x_N\}$, the weighted number X can be calculated using Mixed-Radix Conversion (MRC) formula and the last required coefficient (α_{N-1}) . As α_j s are independent of each other in the process of calculating the values and coefficients of MRC, calculations can be done completely in parallel. On the other hand, [16] tried to find the weighted number equivalent of a residue number consisting of four moduli using CRT and CRT-II algorithms to detect the sign. In this algorithm, each of four residues has a role in the final sign detection. In [17], an algorithm

Manuscript received July 10th, 2021; revised February 5th, 2022.

A. Shahrbanooonezhad is a Ph.D. candidate of the Department of Computer Engineering, Kerman Branch, Islamic Azad University, Kerman 7635168111, Iran (e-mail: alireza_shahrbanooonezhad@yahoo.com).

A.S. Molahosseini is an Assistant Professor of the Department of Computer Engineering, Kerman Branch, Islamic Azad University, Kerman 7635168111, Iran (e-mail: sabbagh@iauk.ac.ir).

A. A. Emrani Zarandi is an Assistant Professor of the Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman 7616913439, Iran (email: a.emrani@uk.ac.ir).

M. Mohammadi is an Associate Professor of the Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman 7616913439, Iran (email: mohammadi@uk.ac.ir).

for general sign detection based on MRC-II and an optimized algorithm for sign detection based on specific moduli set, the last moduli of which must be even, are given. This algorithm, i.e. MRC-II, is similar to MRC and should be calculated parametrically.

This paper solves the problem of RNS sign detection by defining a new representation of signed numbers. Then, an efficient sign detector is proposed which can find sign efficiently based on the proposed representation.

The remainder of this paper is organized as follows: In section II, the concepts of RNS will be explained and dynamic range partitioning will be described for displaying positive and negative numbers. In section III, the proposed representation for the new dynamic range partitioning will be explained. Besides, the proposed hardware architecture and its performance evaluation based on experimental results are presented in section IV.

II. BACKGROUND

RNS is a non-weighted number system that is defined based on the moduli set $\{m_1, m_2, \dots, m_n\}$ that includes co-prime numbers. The representable numbers in any residue number system fall into the range of $[0, M - 1]$, in which $M=m_1.m_2.\dots.m_n$ and is called dynamic range. Each number in RNS has a specific representation, displayed as (x_1, x_2, \dots, x_n) where x_i the smallest positive residue of X divided by m_i and can be showed as $x_i=|X|_{m_i}$. In addition, in order to convert the residue representation of a number (x_1, x_2, \dots, x_n) into an integer, a reverse converter is needed. The Chinese remainder theorem can be used as follows [18]:

$$X = \left| \sum_{i=1}^n M_i \left| m_i^{-1} \right|_{m_i} x_i \right| M \quad (1)$$

where $M_i=M/m_i$ and $|m_i^{-1}|_{m_i}$ is the multiplicative inverse of $|M_i|_{m_i}$. To avoid the large moduli operation for CRT calculations, a new CRT (CRT-II) can be used as below [19]:

$$x_{ij} = x_i + m_i \left| m_i^{-1} \right|_{m_j} (x_j - x_i) \Big|_{m_j} \quad (2)$$

As can be seen, in CRT II each time pairs of (x_i, x_j) and their corresponding moduli (m_i, m_j) can be used, so that $i \neq j$. Therefore X_{ij} is created and this continues until the final number X is generated. Mixed-radix conversion is another technique used for designing reverse converters. In MRC, assuming having a moduli set such as $\{m_1, m_2, m_3\}$ and residue numbers such as (x_1, x_2, x_3) , the equivalent number X can be obtained as follows [25].

$$X = a_3 m_1 m_2 + a_2 m_1 + a_1 \quad (3)$$

Where a_i ($i=1, 2, 3$) is as follows:

$$a_1 = x_1 \quad (4)$$

$$a_2 = \left((x_2 - a_1) \left(\frac{1}{m_1} \right)_{m_2} \right)_{m_2} \quad (5)$$

$$a_3 = \left(\left(\left((x_3 - a_1) \left(\frac{1}{m_1} \right)_{m_3} \right) - a_2 \right) \left(\frac{1}{m_2} \right)_{m_3} \right)_{m_3} \quad (6)$$

Note that, $\alpha = \left(\frac{1}{m_i} \right)_{m_i}$ is multiplicative inverse, where $(\alpha \times m_i) \bmod m_i = 1$. To calculate X , first a_2 and then a_3 are calculated and finally placed in (3).

Signed integers can be represented in RNS, too, for which

at first these numbers must be mapped as unsigned integers. For this purpose the following relation is used:

$$\tilde{X} \equiv \left| x + \left\lfloor \frac{M}{2} \right\rfloor \right|_M - \left\lfloor \frac{M}{2} \right\rfloor \quad (7)$$

\tilde{X} here represents the signed integer. In RNS, the dynamic range is divided into lower and upper half ranges, with the former and the latter displaying positive and negative numbers, respectively. Depending on whether the value of M is even or odd, the division of dynamic range and the exact determination of the range of positive and negative numbers differ; provided that M is even, its residue representation will fall into the range of $[0, (M/2) - 1]$, and when M is odd, it will fall into the range of $[0, (M - 1)/2]$, when $\tilde{x} \geq 0$. On the other hand, its RNS representation will fall into the range of $[M/2, M - 1]$, when $\tilde{x} < 0$ and M is even and when M is odd, it will fall into the range of $[(M + 1)/2, M - 1]$ [20]. In other words, the sign detection of \tilde{X} and its place in dynamic range will be carried out by the examination of the magnitude of the number resulted from (x_1, x_2, \dots, x_n) and according to the cases below, respectively.

when M is even,

$$\text{Sign}(\tilde{X}) = \begin{cases} 0 & \text{if } x \in [0, (M/2) - 1] \\ 1 & \text{if } x \in [M/2, M - 1] \end{cases} \quad (8)$$

when M is odd,

$$\text{Sign}(\tilde{X}) = \begin{cases} 0 & \text{if } x \in [0, (M-1)/2] \\ 1 & \text{if } x \in [(M+1)/2, M-1] \end{cases} \quad (9)$$

Some moduli in RNS such as $2^n - 1$ can ease modular calculation results in an efficient hardware implementation.

Property 1:

The multiplication of an n -bit number X by 2 to the power of k in the moduli $2^n - 1$ can be computed by k bit circular left shift of X [21].

$$|2^k x|_{2^n-1} = \text{CLS}_n(X, K) \quad (10)$$

Property 2:

This feature is a result of Property 1 [7].

$$\begin{aligned} |-2^k X|_{2^n-1} &= |2^k(2^n - 1 - X)|_{2^n-1} \\ &= |2^k \tilde{X}|_{2^n-1} = \text{CLS}_n(\tilde{X}, k) \end{aligned} \quad (11)$$

III. THE PROPOSED METHOD

In this section, a new algorithm is presented for the sign detection of signed integers in RNS. The proposed algorithm consists of two parts: the first part determines the range of positive and negative numbers in dynamic range and the second part expresses the algorithm of sign detection on the new dynamic range partitioning.

A. New dynamic range partitioning

As it was explained in part 2, in all existing residue number systems which can work with signed integers, the dynamic range falls into two lower and upper half ranges that respectively display positive numbers and negative numbers (Fig. 1).

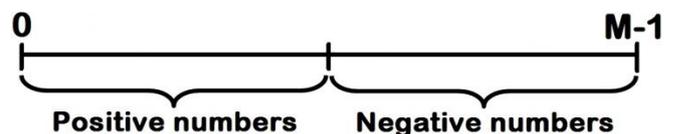


Fig. 1. Conventional dynamic range splitting.

In this partitioning as there is an even moduli in most moduli sets, the value of M is even and the dynamic range is divided into two equal halves. Assuming that there is a moduli set of (m_1, m_2, \dots, m_N) , this paper now divides dynamic range into m_N parts in each of which there will be $S=m_1.m_2.\dots.m_{N-1}$ numbers. Now considering that m_N is one of the odd moduli of the set, the value of S will certainly yield an even value (since one of its covered moduli is even), which will surely allow its division into two equal parts. After the partitioning is done, each S -member set is divided into two equal parts the lower part of which displays positive numbers and the upper part displays negative numbers (Fig. 2).

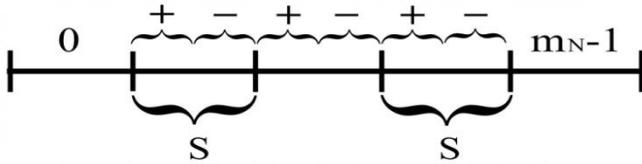


Fig. 2. Suggested dynamic range splitting.

In the light of addition being one of the most important and widely used arithmetic operations required for most calculative applications, we've examined whether and how it is possible to do this operation in the new dynamic range partitioning, with the conducted investigations showing that it is possible to do addition in all three possible modes (addition of two positive numbers, the addition of two negative numbers, the addition of one positive number with one negative number) and based on a certain model. The major advantages of this partitioning for the sign detection are:

1. Lack of mid-range in the process of sign detection

In the common process of dynamic range partitioning and its division into a number of subcategories, there will be a mid-range for recalculations and to determine the exact place of the number for the sign detection. However, in the new partitioning presented in this paper and given that each subcategory will have an independent role in sign detection, there will be no difference between the numbers existing in the mid-range and those in other subcategories, as a result there will be no need to do extra calculations if the examined number falls into any of these subcategories.

2. Doing the operation of sign detection without the need for all moduli and residue numbers

In all sign detection algorithms which suggested for RNS whose dynamic range is divided into two parts, all residues and their corresponding moduli have been used for sign detection [2]-[7]. But using the new dynamic range partitioning, in a RNS with an assumed set of 4 moduli $\{m_1, m_2, m_3, m_4\}$ and 4 residue numbers (x_1, x_2, x_3, x_4) it is possible to use only 3 moduli and 3 residue numbers for the sign detection of the examined number, which leads to a large reduction in calculations and subsequently a simpler hardware implementation. How this algorithm is done and calculated is explained in the next section.

B. New sign detection algorithm

In all common sign detection algorithms, all moduli of an examined set and their corresponding residue numbers are used for the sign detection, which causes an increase in the required calculations, hardware capacity and thus a delay in sign detection.

This section deals with the new sign detection algorithm, and shows that in a 4-moduli set, the sign can be detected using 3 moduli and 3 residue numbers. The examined moduli set is $\{2^{2n}, 2^n-1, 2^n+1, 2^{n+1}-1\}$ where $m_1=2^{2n}$, $m_2=2^n-1$, $m_3=2^n+1$, $m_4=2^{n+1}-1$, and the residue numbers are (x_1, x_2, x_3, x_4) . Regarding methods of sign detection, this algorithm is based on partial reverse conversion. This method uses dynamic range, classifying it into m_4 ranges each of which will have $S=m_1.m_2.m_3$ numbers. Now that there is a moduli with an even value, S also has an even value that can be divide into two equal parts. In this method first two pairs of (m_2, m_3) and (x_2, x_3) are used and then using CRT a residue number equal to (P_x) and a new moduli equal to $(DR_2=m_2.m_3)$ will be calculated.

Equations for calculating P_x

As mentioned earlier, using CRT and two pairs of (m_2, m_3) and (x_2, x_3) P_x can be calculated as follows:

$$P_x = |X|_{m_2.m_3} \quad (12)$$

$$= \left| m_3 \cdot \left| m_3^{-1} \right|_2 \cdot x_2 + m_2 \cdot \left| m_2^{-1} \right|_{m_3} \cdot x_3 \right|_{m_2.m_3}$$

Which $\left| m_2^{-1} \right|_{m_3} = \left| m_3^{-1} \right|_{m_2} = 2^{n-1}$

Proof: $\left| m_3^{-1} \right|_{m_2} = 2^{n-1}$

$$\begin{aligned} \left| m_3^{-1} \right|_{m_2} &= |(2^n + 1)^{-1}|_{2^{n-1}} \\ &= |k \times (2^n + 1)|_{2^{n-1}} \\ &= |k \times 2|_{2^{n-1}} = 1 \rightarrow k = 2^{n-1} \end{aligned}$$

Proof: $\left| m_2^{-1} \right|_{m_3} = 2^{n-1}$

$$\begin{aligned} \left| m_2^{-1} \right|_{m_3} &= |(2^n - 1)^{-1}|_{2^{n+1}} \\ &= |(2^n + 1 - 1 - 1) \times 2^{n-1}|_{2^{n+1}} \\ &= |-2 \times 2^{n-1}|_{2^{n+1}} \\ &= |-(2^n + 1 - 1)|_{2^{n+1}} = 1 \rightarrow k = 2^{n-1} \end{aligned}$$

Equations for calculating Q_x

Assuming that $DR_2=m_2.m_3$ and $S=DR_2.m_1$ and thus having two pairs of (P_x, x_1) and (DR_2, m_1) , Q_x can be calculated using CRT II. This section will show that Q_x will determine the exact place of the number X in a subcategory of dynamic range partitioning.

$$Q_x = x_1 + m_1 \left| \left| m_1^{-1} \right|_{DR_2} \cdot (P_x - x_1) \right|_{DR_2} \quad (13)$$

which will be $\left| m_1^{-1} \right|_{DR_2} = 1$.

It should be noted that $\left| (2^{2n})^{-1} \right|_{2^{2n-1}} = 1$ since $\left| (2^{2n}) \times 1 \right|_{2^{2n-1}} = |2^{2n} + 1 - 1|_{2^{2n-1}} = 1 \rightarrow k = 1$.

Now three moduli and three residue numbers have been used, the value of Q_x has been obtained and we know that the final value of X can be calculated using two pairs of (Q_x, x_4) and (S, m_4) and CRT II as below:

$$X = \left| \left| S^{-1} \right|_{m_4} \cdot (x_4 - Q_x) \right|_{m_4} \cdot S + Q_x \quad (14)$$

Thus by comparing the above expression with $X=R_x.S+Q_x$, we will realize that Q_x is an indicator of the

place of the examined number in its related subcategory, which, considering the dynamic range partitioning explained above, allows determining the examined number using Q_x without inserting the values of m_4 and x_4 into calculations, because as shown in (14), m_4 and x_4 are used for determining exactly the number of the examined number's subcategory, and how negative numbers are represented in the new dynamic range partitioning. It should be noted that there will be no need to know the place and number of the examined number's subcategory and to insert m_4 and x_4 into the calculations for the sign detection (Fig. 3). This will significantly reduce delay in sign detection and decrease the required hardware.

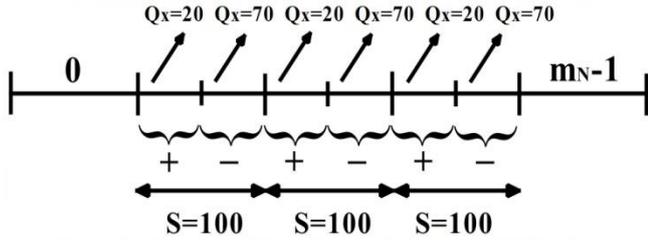


Fig. 3. Determining the sign of number using the same Q_x .

As the figure above shows, a number with $Q_x=20$ will certainly be a positive number and one with $Q_x=70$ will surely be a negative one. Without knowing the exact number of its subcategory (that is, without inserting m_4 and x_4 into the calculations and determining the value of R_x); with the calculation of Q_x and comparing it to S , sign detection can be done. In summary:

$$\text{Sign}(x) = \begin{cases} 1 & \text{if } Q_x \geq S/2 \\ 0 & \text{if } Q_x < S/2 \end{cases} \quad (15)$$

C. Numerical example

In the moduli set $\{2^{2n}, 2^{n-1}, 2^{n+1}, 2^{n+1}-1\}$ and assuming $n=2$, we will have $m_1=16, m_2=3, m_3=5$ and $m_4=7$ which $S=m_1 \cdot m_2 \cdot m_3=240$. The full dynamic range consists of 7 partitions where each one includes 240 numbers.

According to the partitioning done on the dynamic range, it is clear that the number $X=-300$ will be equal to 660 on the dynamic range and as $S=240$, the value of $X=660$ will be calculated based on the formula $X=R_x \times S + Q_x$ as below:

As $660=2 \times 240+180$, $Q_x=180$, based on the value of $S=240$ it will be clear that the number 660 with $Q_x=180$ will fall into the second half of the related subcategory and will be a negative number (Fig. 4).

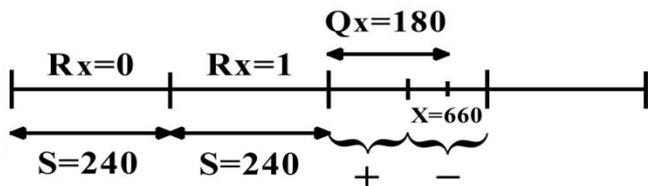


Fig. 4. Numerical example.

In the next section and after describing the necessary hardware units for implementing the above equations, the numerical example on the presented hardware will be described with detail.

D. The proposed hardware architecture

The proposed architecture for the sign detection can be divided into three general parts. The first and second parts use 3 residue numbers to produce Q_x and P_x and the third part uses the produced Q_x and compares it to $S/2$ to determine the sign for the examined number (Fig. 5).

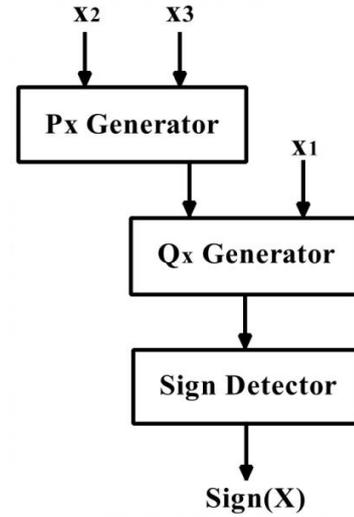


Fig. 5. Architecture of the proposed algorithm.

The hardware needed for P_x can be designed and implemented by placing $|m_2^{-1}|_{m_3}=|m_3^{-1}|_{m_2}=2^{n-1}$ in (12) and calculating its final equation.

$$\begin{aligned} P_x &= \left| m_3 \cdot \left| m_3^{-1} \right|_{m_2} \cdot x_2 + m_2 \cdot \left| m_2^{-1} \right|_{m_3} \cdot x_3 \right|_{m_2 \cdot m_3} \quad (16) \\ &= |(2^n + 1)(2^{n-1})x_2 + (2^n - 1)(2^{n-1})x_3|_{2^{2n-1}} \\ &= |2^{2n-1}x_2 + 2^{2n-1}x_2 - 2^{n-1}x_3 + 2^{2n-1}x_3|_{2^{2n-1}} \end{aligned}$$

The above expression results in two independent terms called K_1 and K_2 based on the factors X_2 and X_3 . Let $K_1 = |2^{2n-1}x_2 + 2^{n-1}x_2|_{2^{2n-1}}$ and $K_2 = |-2^{n-1}x_3 + 2^{2n-1}x_3|_{2^{2n-1}}$.

Now following property 1 and property 2 as previously presented, K_1 and K_2 can be calculated exactly.

$K_1 = |CLS_{2n}(x_2, 2n-1) + CLS_{2n}(x_2, n-1)|_{2^{2n-1}}$ (17) Besides, since x_2 has n bits, which must increase to $2n$ bits according to the (17), n -bit zeros will be added to its left side and then a circular left shift will be applied.

$$\begin{aligned} K_1 &= |x_{2,0} \underbrace{00 \dots 00}_{n \text{ bit}} x_{2,n-1} \dots x_{2,1} + \\ &0 \underbrace{x_{2,n-1} \dots x_{2,0}}_{n \text{ bit}} \underbrace{00 \dots 00}_{n-1 \text{ bit}} |_{2^{2n-1}} \\ &= x_{2,0} \underbrace{x_{2,n-1} \dots x_{2,0}}_{n \text{ bit}} \underbrace{x_{2,n-1} \dots x_{2,1}}_{n-1 \text{ bit}} \end{aligned}$$

Moreover:

$$K_2 = |CLS_{2n}(x_3, 2n-1) + CLS_{2n}(x_3, n-1)|_{2^{2n-1}} \quad (18)$$

Since x_3 has $n+1$ bit and must increase to $2n$ bits according to the (18), $(n-1)$ zero bits will be added to its left side and circular left shift will be applied.

$$K_2 = |x_{3,0} \underbrace{00 \dots 00}_{n-1 \text{ bit}} \underbrace{x_{3,n} \dots x_{3,1}}_{n \text{ bit}} + \underbrace{\bar{x}_{3,n} \dots \bar{x}_{3,0}}_{n+1 \text{ bit}} \underbrace{11 \dots 11}_{n-1 \text{ bit}} |_{2^{2n-1}}$$

Now by examining x_3 bits and given that when its $(n+1)$ bit has a value of 1, the rest of its bits will be zero and while $x_{3,n} = 1$, the above expression can be added and a constant

value can be obtained as:

$$L = \underbrace{00\dots00}_n \underbrace{100\dots00}_{n-1} + 0 \underbrace{11\dots11}_{2n-1}$$

$$= 1 \underbrace{00\dots00}_n \underbrace{11\dots11}_{n-1}$$

Thus based on the value of $x_{3,n}$ bit, the expression P_x can be defined as a bi-conditional expression:

$$P_x = \begin{cases} |K_1 + L|_{2^{2n-1}} & \text{if } x_{3,n} = 1 \\ |K_1 + K_3 + L|_{2^{2n-1}} & \text{if } x_{3,n} = 0 \end{cases} \quad (19)$$

Where $K_3 = x_{3,0} \bar{x}_{3,n-1} \dots \bar{x}_{3,0} x_{3,n-1} \dots x_{3,1}$.

As the above calculations and results show, there is no need for a complex hardware architecture. The proposed hardware implements the above expression and using only bit inversion and reshuffling, the concerned results can be obtained and then reach the value of P_x , and n modified full adders (MFA) can be used [21] where each of them includes two modified half adders called MHA1 and MHA2 and its internal structure is shown in Fig. 6. Fig. 6 shows that the inputs of MFA are x_2 and x_3 bits. Using MHA1 a range of vectors including $x_{3,i}$ and $x_{2,i}$ per ($i \in [1, n - 1]$), a constant vector with the value of 1 can be accumulated. Similarly, using MHA2, two vectors of $x_{2,i}$ and $(\bar{x}_{3,i} \cdot \bar{x}_{3,n})$ per $i \in [0, n - 1]$ will be accumulated. Finally, MFAs with a little sorting will generate two $2n$ -bit vectors of Sum and Carry called g and f which will be used for calculating Q_x .

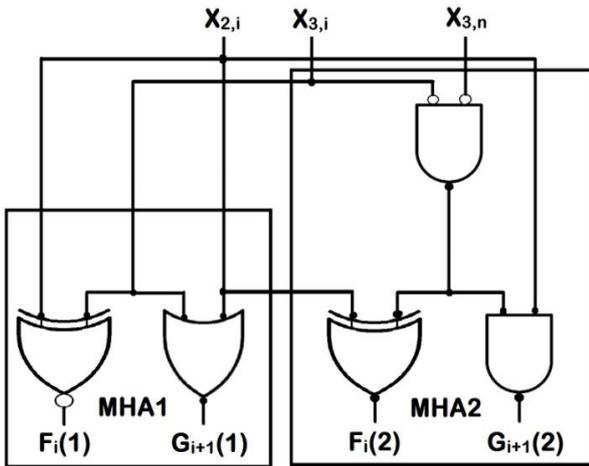


Fig. 6. Circuit architecture of MFA.

Now by placing $m_1 = 2^{2n}$, $|m_1^{-1}|_{DR2} = 1$ and $P_x = |f + g|_{2^{2n-1}}$ in (13), the value of Q_x is obtained as below:

$$Q_x = |1 \times (f + g - x_1)|_{2^{2n-1}} \cdot 2^{2n} + x_1 = x_1 + 2^{2n} \cdot Z \quad (20)$$

Comparing the two sides of the equation will give:

$$Z = |1 \times (f + g - x_1)|_{2^{2n-1}} \quad (21)$$

$$= |f + g - x_1|_{2^{2n-1}}$$

$$= |z_1 + z_2 + z_3|_{2^{2n-1}}$$

As it was shown, the factor Z is obtained from the modulo sum of 3 vectors to the moduli $2^{2n} - 1$, where $z_1 + z_2 + z_3$ is calculated as below:

$$z_1 = f \quad (22)$$

$$z_1 = f_{2n-1} \dots \dots \dots f_0 \quad (23)$$

$$z_2 = g \quad (24)$$

$$z_2 = g_{2n-1} \dots \dots \dots g_0$$

$$z_3 = -x_1 = \bar{x}_1$$

$$z_3 = \bar{x}_{1,2n-1} \dots \dots \dots \bar{x}_{1,0}$$

Now according to (21), the final value of Z can be calculated using a carry-save adder (CSA) and a modular adder by the moduli $2^{2n} - 1$. On the other hand, as $x_1 \in [0, 2^{2n})$ and will have $2n$ -bits, and in the (20), Z has a coefficient of 2^{2n} , it is possible to calculate the value of Q_x without extra hardware with the help of concatenation as below (the hardware implementation is shown in Fig. 7).

$$Q_x = \underbrace{Z_{2n-1} \dots \dots \dots Z_0}_{2n \text{ bit}} \underbrace{X_{1,2n-1} \dots \dots \dots X_{1,0}}_{2n \text{ bit}}$$

After calculating Q_x as previously explained, its comparison with the value of $S/2$ will allow us to determine the examined sign; if Q_x is in the lower half of S , the number will be positive and if it is in the upper half, it will be negative (Fig. 8).

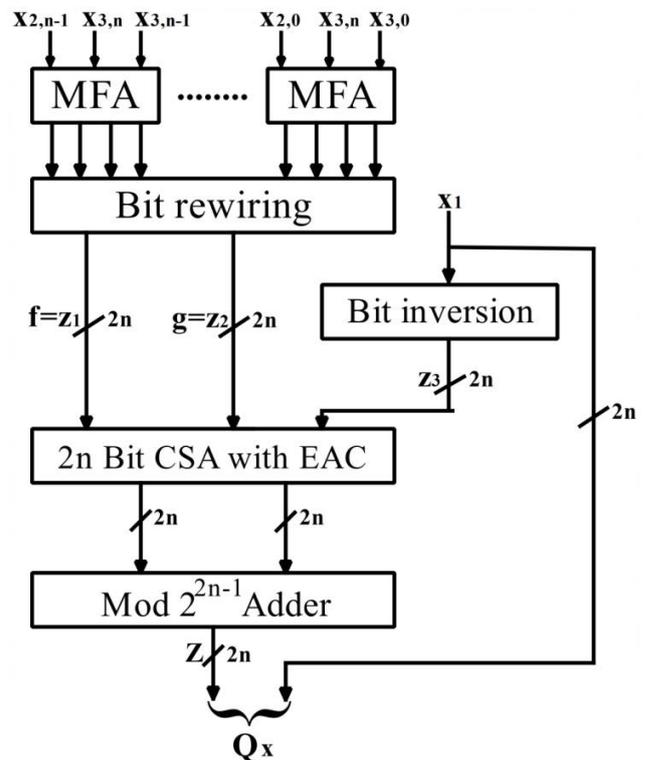


Fig. 7. Circuit architecture of Q_x generator.

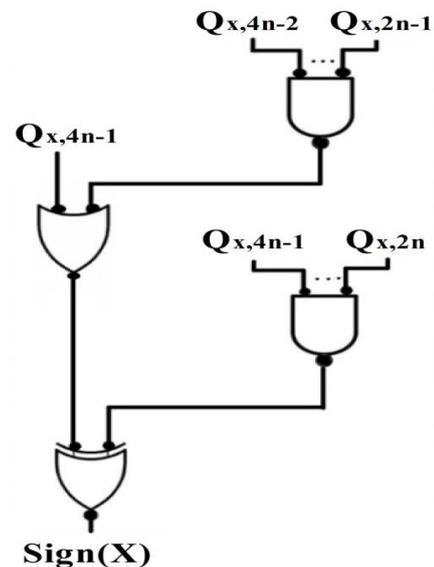


Fig. 8. Hardware implementation of sign detector unit.

After calculating Q_x as previously explained, its comparison with the value of $S/2$ will allow us to determine the examined sign; if Q_x is in the lower half of S , the number will be positive and if it is in the upper half, it will be negative (Fig. 8).

E. Example

Consider the moduli $m_1=2^{2n}$, $m_2=2^n-1$, $m_3=2^{n+1}$ and $m_4=2^{n+1}-1$, and assuming $n=2$, we will have: $m_1=16$, $m_2=3$, $m_3=5$ and $m_4=7$. Now with the aim of investigating the sign of the number $X=-300$ and the division of dynamic range according to Fig. 4, we must address the sign of the number 660 that is equal to -300 . After computing the residues, we will have: $x_1=4=0100$, $x_2=0=00$, $x_3=0=000$ and $x_4=2=010$, and by applying them to Fig. 7, the calculations will be done.

Given the hypothesis $n=2$, there will be two numbers of MFA, which considering the values of x_2 and x_3 bits, two 2-n-bit vectors with the values of (1111) and (0000) that are respectively called f and g , will be generated. These vectors along with the vector $Z_3=\bar{x}_1=(1011)$ will act as the inputs of 2n-bit CSA with EAC. Next, the outputs of sum and carry will have the values of (1011) and (0100) respectively and will enter the modular adder as the inputs of the next level so that the final value of $P_x=1011$ is obtained. As the Fig. 7 shows, the value of the final output of Q_x will be obtained from the concatenation of P_x and x_1 and be equal to (10110100). As the final output shows, $Q_x=10110100=180$ which, with the calculation of the value Q_x , will be equal to the numerical example mentioned previously. Now considering Fig. 8, the negativity of the investigated number will be clear by applying the value Q_x to this circuit and that the bit $MSB=1$.

IV. PERFORMANCE EVALUATION

In this section, the performance proposed algorithm is evaluated based on the 4-moduli set $\{2^{2n}, 2^n-1, 2^{n+1}, 2^{n+1}-1\}$ and the obtained results are compared with one of the latest existing algorithms presented in [7] when $k=n$.

To compare the results, at first, the codes related to producing a hardware model of sign detection in [7] and the proposed algorithm are written using VHDL, and then their performance is evaluated and confirmed using Modelsim. The written codes are assessed on the above moduli set and for $n=4,8,10,12,16,20$, and then every design is synthesized using Synopsys design compiler version C-2009.06 using SC_TSMC 180 nm technology standard cell library. The hardware efficiency between the proposed design and [7] can also be analyzed at a component level. The comparison of the hardware gates required for the proposed work and the algorithm in [7], as shown in Table I, reveal that in the proposed design with a decline by half in the hardware required for [7], we will expect a 50% reduction in some factors of the circuit such as area and critical path delay, with the results of the synthesis demonstrating it well.

The measured area in μm^2 and the critical path delay in ns of each design are obtained for a different n and plotted in Figs. 9 and 10. In addition, the total power consumption in mW and the leakage power in μW are plotted in Figs. 11 and 12.

TABLE I
EVALUATION OF RNS SIGN DETECTION ARCHITECTURE

Item	[7]	Proposed
MFA	Yes	Yes
Bit rewiring	2 levels	1 level
Bit inversion	Yes	Yes
Circular left shift	1 level	No
2n bit CSA with EAC	1 level	1 level
Mod $2^{2n}-1$ adder	1 level	1 level
(n+1)bit CSA with EAC	2 or 3 levels	No
Circular right shift	1 level	No
CSA with EAC tree	Yes	No
Mod $2^{n+1}-1$ adder	1 level	No

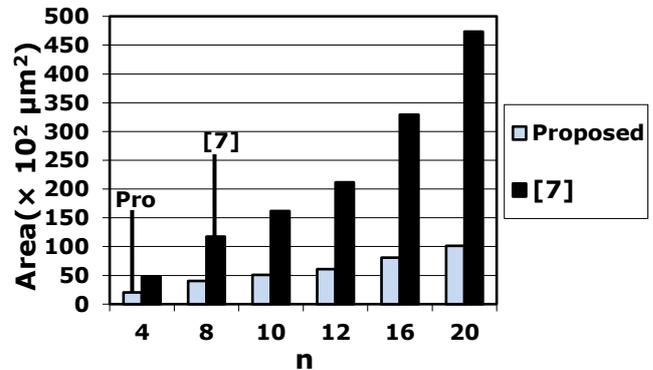


Fig. 9. Comparison of area of the proposed design with the sign detector based on [7] with $n=4, 8, 10, 12, 16, 20$.

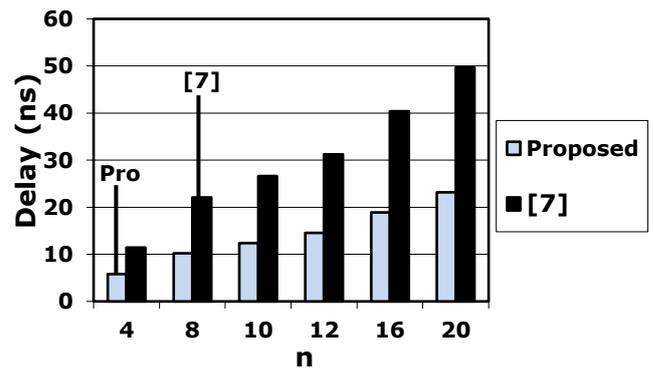


Fig. 10. Comparison of delay of the proposed design with the sign detector based on [7] with $n=4, 8, 10, 12, 16, 20$.

Fig. 9 shows the proposed design is 69% smaller than the design presented in [7] on average. Also according to the information given in Fig. 10, the average critical path delay in the proposed design is 52.5% better than the one in [7].

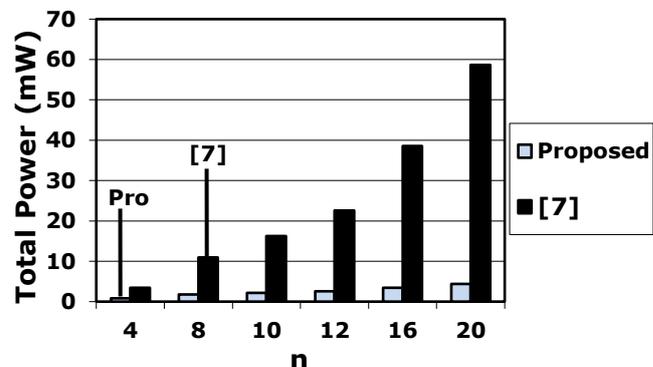


Fig. 11. Comparison of total power of the proposed design with the sign detector based on [7] with $n=4, 8, 10, 12, 16, 20$.

Figs. 11 and 12 show reductions in the average power consumption and the average leakage power by 86.2% and

70.4%, respectively. Besides, Figs. 9, 11 and 12 show that the rate of improvement in all three parameters of area, power consumption and leakage power rises with the increase in the value of n .

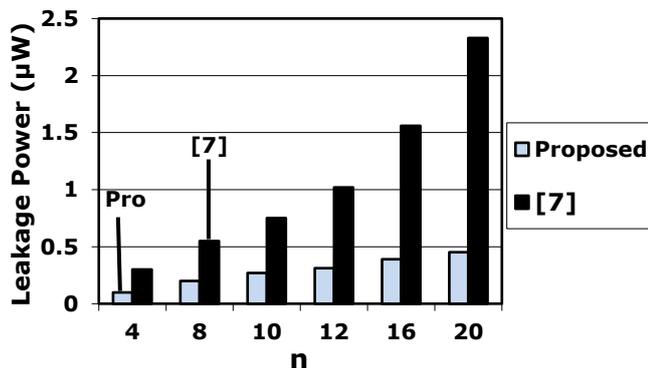


Fig. 12. Comparison of leakage power of the proposed design with the sign detector based on [7] with $n=4, 8, 10, 12, 16, 20$.

As explained earlier, whatever the number of moduli is greater and the dynamic range is larger, then the number of numbers that can be displayed in that system will be higher.

In the following, we compare the proposed algorithm with dynamic range $5n$ to each one of the presented methods in [15], [17] and [22], that each of them has a dynamic range of $3n$. As shown in Figs. 13 and 14, the results of the proposed algorithm in terms of area and power are better than the results of [15], [17] and [22], which shows the superiority of the proposed algorithm.

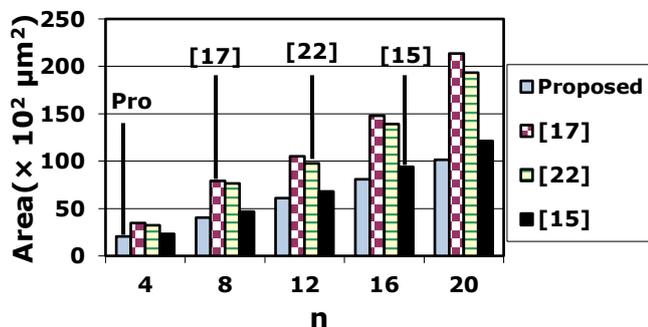


Fig. 13. Comparison of area of the proposed design with the sign detectors based on [15], [17] and [22] with $n=4, 8, 12, 16, 20$.

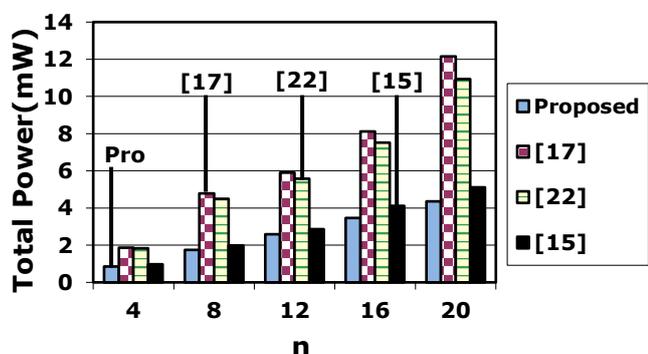


Fig. 14. Comparison of total power of the proposed design with the sign detectors based on [15], [17] and [22] with $n=4, 8, 12, 16, 20$.

V. CONCLUSION

This paper presents a new method for the effective sign detection on the 4-moduli set $\{2^{2n}, 2^{n-1}, 2^{n+1}, 2^{n+1}-1\}$. In the proposed algorithm, the dynamic range partitioning has been changed than previous works that allowed us to use only 3 residue numbers and 3 moduli out of 4 residue numbers and 4 moduli to determine the sign of the examined number. Given that one residue number and one moduli do not enter the computations of the sign detection, a considerable decrease in the parameters of the circuits is achieved. The experimental results show that on average 69%, 52.5%, 86.2% reductions are obtained in area, critical path delay, and power consumption.

REFERENCES

- [1] C. H. Chang, A. S. Molahosseini, A. A. E. Zrandi, and T. F. Tay, "Residue number system: 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, 2015.
- [2] S. Kumar and C. H. Chang, "A scaling-assisted signed integer comparator for the balanced five-moduli set RNS $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1, 2^{n+1} - 1\}$," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 12, pp. 3521 - 3533, 2017.
- [3] J. Alves Jr., L. F. L. Nascimento, and L. C. P. Albini, "Using the redundant residue number system to increase routing dependability on mobile ad hoc networks," *J. Sel. Areas Telecommun.*, vol. 2, pp. 67-73, 2011.
- [4] T. Tomczak, "Fast sign detection for RNS $\{2^n - 1, 2^n, 2^n + 1\}$," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 6, pp. 1502-1511, 2008.
- [5] N. Samimi, M. Kamal, A. Afzalli-kusha, and M. Pedram, "Res-DNN: A residue number system based DNN accelerator unit," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 2, pp. 658-671, 2020.
- [6] L. Sousa, R. Paludo, P. Martins, and H. Pettenghi, "Toward the integration of reverse converters into the RNS channels," *IEEE Trans. Comput.*, vol. 69, no. 3, pp. 342-348, 2020.
- [7] S. Kumar, C. H. Chang, and T. F. Tay, "New algorithm for signed integer comparison in $\{2^{n+k}, 2^{n-1}, 2^{n+1}, 2^{n+1}-1\}$ and its efficient hardware implementation," *IEEE Trans. Circuits Syst. I*, vol. 64, no. 6, pp. 1481-1493, 2016.
- [8] L. Ronghui and Z. Jingpu, "Application of Deep Extreme Learning Machine in Network Intrusion Detection Systems," *IAENG International Journal of Computer Science*, vol. 47, no. 2, pp. 136-143, 2020.
- [9] D. D. Miller, R. E. Altschul, J. R. King, and J. N. Polky, "Analysis of the residue class core function of Akushskii, Burcev and Pak," *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*, 1986, New York, USA, pp. 390-401.
- [10] M. Lu and J. Chiang, "A novel division algorithm for the residue number system," *IEEE Trans. Comput.*, vol. 41, no. 8, pp. 1026-1032, 1992.
- [11] Y. Wang, X. Song, and M. Aboulhamid, "A new algorithm for RNS magnitude comparison based on new chinese remainder theorem II," in *Proc. 1999 9th Great Lakes Symp. VLSI*, pp. 362-365, 1999.
- [12] L. Sousa and P. Martins, "Efficient sign identification engines for integers represented in RNS extended 3-moduli set $\{2^{n-1}, 2^{n+k}, 2^{n+1}\}$," *Electron. Lett.*, vol. 50, no. 16, pp. 1138-1139, 2014.
- [13] L. Sousa and P. Martins, "Sign detection and number comparison on RNS 3-moduli sets $\{2^{n-1}, 2^{n+k}, 2^{n+1}\}$," *Circuits Syst. Signal Process.*, vol. 36, no. 3, pp. 1224-1246, 2017.
- [14] Z. Wang, G. A. Jullien, and W. C. Miller, "An improved residue-to-binary converter," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 47, no. 9, pp. 1437-1440, 2000.
- [15] M. Xu, Z. Bian, and R. Yao, "Fast sign detection algorithm for the RNS moduli set $\{2^{n+1}-1, 2^{n-1}, 2^n\}$," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 2, pp. 379-383, 2015.
- [16] C. H. Chang and S. Kumar, "Area-efficient and fast sign detection for four-moduli set RNS $\{2^{n-1}, 2^n, 2^{n+1}, 2^{2n+1}\}$," in *Proc. 2014 IEEE Int. Symp. Circuits Syst.*, 2019 Melbourne, Australia, pp. 1540-1543.
- [17] M. Akkal and P. Siy, "Optimum RNS sign detection algorithm using MRC-II with special moduli set," *J. Syst. Arch.*, vol. 54, no. 10, pp. 911-918, 2008.

- [18] N. S. Szabo and R. I. Tanaka, "Residue Arithmetic and Its Applications to Computer Technology," New York: Mc Grow-Hill, 1967.
- [19] Y. Wang, "Residue-to-binary converters based on new Chinese remainder theorems," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 3, pp. 197-205, 2000.
- [20] T. F. Tay, C. H. Chang, and J. Y. S. Low, "Efficient VLSI implementation of 2^n scaling of signed integer in RNS $\{2^n-1, 2^n, 2^{n+1}\}$," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 10, pp. 1936-1940, 2013.
- [21] T. F. Tay, C. H. Chang, and L. Sousa, "Base transformation with injective residue mapping for dynamic range reduction in RNS," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 9, pp. 2248-2259, 2015.
- [22] E. A. Radadi and P. Siy, "RNS sign detector based on Chinese remainder theorem II (CRT II)," *Computers & Mathematics with Applications.*, vol. 46, no. 10, pp. 1559-1570, 2003.
- [23] H. Bronnimann, I. Z. Emiris, V. Y. Pan, and S. Pion, "Computing exact geometric predicates using modular arithmetic with single precision," in *Proc. 13th Annu. Symp. Comput. Geometry*, 1997, Nice, France, pp. 174-182.
- [24] L. Sousa, "Efficient method for magnitude comparison in RNS based on two pairs of conjugate moduli," in *proc. 2007 18th IEEE Symp. Comput. Arithmetic*, 2007, Montpellier, France, pp. 240-250.
- [25] P. V. Ananda-Mohan and P. S. Phalguna, "Evaluation of Mixed-radix digit computation techniques for the three moduli RNS $\{2^n-1, 2^n, 2^{n+1}-1\}$," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 68, no. 4, pp. 1418-1422, 2021.
- [26] S. Kumar and C. H. Chang, "A new fast and area efficient adder-based sign detector for RNS $\{2^n-1, 2^n, 2^{n+1}\}$," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 7, pp. 2608-2612, 2016.