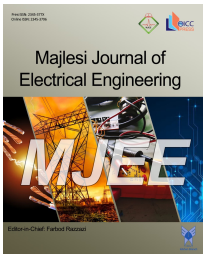

Volume 18, Issue 2, 182429 (1-15)

Majlesi Journal of Electrical Engineering (MJEE)

https://dx.doi.org/10.57647/j.mjee.2024.1802.29

A hybrid forward/reverse converter in reversible logic to reduce hardware complexity of residual number system

Ailin Asadpour[1], Amir Sabbagh[1,*], Azadeh Emrani[2]

[1]*Department of Computer Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran.*
[2]*Department of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, Iran.*

[*]**Corresponding author:** sabbagh@iauk.ac.ir

Original Research

Received:
4 December 2023
Revised:
14 March 2024
Accepted:
28 March 2024
Published online:
11 May 2024

Abstract:

As an emerging technology, reversible computing enables the development of high-performance computing systems with low energy consumption. A residual number system (RNS) that performs arithmetic operations in parallel with error tolerance and no carry propagation requires forward and reverse converters to communicate with other digital circuits. Designing reversible forward and reverse converters using new technologies is very important due to their wide applications in implementing the RNS. These converters, which are the overhead of the system, increase energy consumption. This study proposes a hybrid converter conforming to reversible logic for the RNS. This hybrid converter unifies forward and reverse converters by sharing hardware and reversible gates. By using the mixed-radix conversion (MRC), the reverse conversion arithmetic relations adopt a similar format to that of the forward conversion arithmetic relations, and by adding a number of Fredkin gates and modifying the inputs, the reverse converter hardware is used to perform forward conversion. Based on the findings, the hybrid converter, which conformed to reversible logic for the moduli set $\{2^{2n}, 2^n-1, 2^{n+1}-1\}$ and $\{2^n-1, 2^n+1, 2^{2n}+1\}$, decreased the quantum cost to 19.56% and 19.52%, respectively.

Keywords: Computer arithmetic; Arithmetic digital circuits; Residue number system (RNS); Forward converter; Reverse converter; Moduli adder

1. Introduction

The embedded digital signal processing systems used currently require the use of minimum energy computing structures in hardware implementations of modern applications such as deep learning and cryptography [1–3]. For hardware implementation of applications such as convolutional neural networks, which have a very large number of additions and multiplications, the design of special structures with minimum energy consumption is necessary [1]. Also, due to the reduction of power consumption and the increase in the speed of computing circuits, the residual number system (RNS) is the most common special-purpose numerical system, which is widely used in the hardware implementation of computing algorithms based on addition, subtraction, and multiplication with limited and in-scale carry propagation, including digital signal processing, and deep convolutional neural networks [2, 4–6]. On the other hand, contrary to what Moore had predicted about the development of semiconductors, even modern transistors are not significantly more efficient than earlier ones [7]. Moreover, new technologies such as quantum circuit technology have been the subject of wide investigations on the design and processing of systems with ultra-low power consumption. Quantum technology has many applications in optical computing and Quantum-dot Cellular Automata (QCA), and one of its most important applications includes reversible computing based on reversible logic, which has the minimum energy loss and prevents information loss [8]. Since adders and multipliers are widely used in RNSs, combining the features of RNS with the structure of reversible circuits enables developing a new design in digital circuits [9]. In this article, drawing on the main components of the RNS, including the forward converter and the inverse converter, which convert normal weighted binary numbers into the residual represented number and vice versa, and given that both the for-

ward and inverse converters are made based on carry-save adders (CSAs) and modular adders, a new computing unit based on reversible logic is proposed, which can implement both forward and inverse conversions. The most important advantage of the proposed design is that the overhead is significantly reduced as there are two separate designs for the forward and reverse converters. In the hybrid converter proposed, a single hardware generates both forward conversion output (residuals) and reverse conversion output (binary weighting number, which is handled by a control signal). To achieve this goal using an alternative approach, the new Chinese residual theorem (CRT) algorithm was used, which converts the reverse conversion calculation relations into forward conversion calculation relations.
Then, based on multiple multiplexers, the hardware will be shared; thus, the correct input based on the control signal is selected and applied to the internal adders of the circuit.
The results show that the use of the proposed circuit results in a significant reduction of the chip area because rather than using two separate converters, a single dual-purpose converter is used. In section 2, the key concepts of RNS and reversible logic will be reviewed. Then, in section 3, direct and inverse conversion formulas for integration will be rewritten. Section 4 presents the proposed design in general, and the structure of the proposed reversible RNS is described. Also, a special case of the proposed design is presented for the modular set $\{2^{2n},\ 2^n-1,\ 2^n+1\}$ and $\{2^n+1,\ 2^n+1,\ 2^n-1\}$. Finally, section 5 concludes the paper, and the key findings are discussed.

2. An overview of the residual numbers system and reversible logic

This section provides an overview of RNS followed by a review of reversible logic and a detailed discussion of key concepts underlying it.

2.1 Residue number system (RNS)

RNS includes a set of pairwise co-prime numbers called moduli set, which is displayed as $m_1, m_2, \ldots, m_n$. Modular multiplication indicates the number and range of numbers that can be displayed in the system, which is called the dynamic range of the RNS and is expressed as $[0, M-1]$ [10].

$$M = m_1 \times m_2 \times \ldots m_{\mathrm{n}} \tag{1}$$

In RNS, first, normal weighted binary numbers are divided by the moduli set using the forward converter.
The obtained remainders are the remainder representation of that number in RNS, which are shown as follows.

$$X \xrightarrow{\text{RNS}} (x_1, x_2, \ldots, x_{\mathrm{n}}) \quad \text{where } x_{\mathrm{i}} = |X|_{m_i} \quad \text{for } i = 1, \ldots, n \tag{2}$$

Then arithmetic operations of moduli addition, subtraction, or multiplication are implemented in parallel and without carry propagation on the remainders, as follows:

$$X \square Y \xrightarrow{\text{RNS}} (|x_1 \square y_1|_{m_1}, |x_2 \square y_2|_{m_2}, \ldots, |x_{\mathrm{n}} \square y_{\mathrm{n}}|_{m_n}) \tag{3}$$

So
$\square\ \varepsilon\{+, -, \times\}$

Finally, the result is a set of residual digits converted to a weighted equivalent value using an inverse converter based on inverse conversion algorithms such as CRT or mixed-radix conversion (MRC) so that they can be understood and processed by the remaining components of the system [11].

2.2 An overview of reversible logic

Reversible calculations are developed as a novel approach to save energy, minimize energy loss, and optimize circuits compared to previous calculation methods. Reversible computing circuits use computing systems with minimum energy consumption and high efficiency. In classical calculations, where energy consumption can be minimized, some parts will remain in the system due to the irreversibility of the calculation. In irreversible logic circuits, the number of circuit inputs outweighs the number of outputs, which causes some bits containing information to be lost during processing, and their electrical energy is released as heat energy [14].
Research on reversible gates dates back to 1960. The motivation behind the studies was the minimum heat (or basically no heat) generation by these gates. The heat generated by irreversible logic calculations per bit of information is equal to $\mathrm{KT} \times \ln 2$, where K is Boltzmann's constant and T is the temperature. At room temperature, the amount of heat consumed is low (2.9×10^{-21} joules) [15]. While this amount seems very insignificant on a small scale, it cannot be overlooked. Considering Moore's law that the number of transistors and elements in digital circuits increases by 100% every 18 months, it is not far from expected that the loss of electrical energy and the resulting heat production will become a serious challenge in the design of large-scale integrated circuits [16]. The consumption of energy in conventional irreversible circuits is due to information loss. In 1993, Bennet argued that by performing reversible calculations, the total energy consumed due to the loss of information in the reversibility process is eliminated [17]. In reversible circuits, thanks to the inverting circuit, the direction of calculations can be reversed so that the system can be returned to its initial state or any point in the calculation history. In a system, calculations are reversible if reversible gates are used. Reversible bits create a one-to-one mapping between input and output.
Information is lost when it is not possible to recover the inputs from the outputs. Thus, the number of inputs and the number of outputs are the same in reversible circuits. In reversible logic, feedback from output to input is not possible and there is no Fan out, so all reversible gates are also invertible, that is, by using an inverted gate, the gate inputs can be retrieved from the gate outputs. The evaluation of quantum circuits is performed with different criteria, defined as follows:
Definition 1: the quantum cost of a reversible gate is equal to the number of quantum gates 1×1 and 2×2, used in the implementation of the reversible gate. The quantum cost of all 1×1 and 2×2 reversible gates equals 1 [18]. To calculate the quantum cost, some gates can be used whose quantum costs were pre-determined.

Definition 2: garbage outputs are unwanted or unused outputs that, as primary output lines, are not always important and are only added to keep the circuit's reversibility [19].
Definition 3: the ancilla input is an input that must be held constant at 0 or 1 for the circuit to operate properly. Garbage outputs and ancilla inputs relate to each other as follows [19]: Input + ancilla input = output + garbage output
Definition 4: hardware complexity refers to the total number of logic operations obtained by calculating the number of AND, NOT, and XOR operations used in the reversible circuit. The following parameters should be considered when determining the hardware complexity of reversible circuits:
α = number of two-input XOR gates
β = number of two-input AND gates
δ = number of NOT gates
Definition 5: the maximum number of gates in a path from each input line to each output line represents the delay of the logic circuit. Based on this definition, calculations are performed at each gate per unit of time, and before the calculations begin, all inputs to the circuit are available.
Other reversible gates can be made using the primary gates. For example, by setting the control inputs to a fixed value of zero, two reversible gates, PG and HNG, can be made, which are used as half adder (HA) and full adder (FA), respectively, in quantum circuits [10, 12, 20–23]. Fig. 1(a) and Fig. 1(b) demonstrate the symbols and quantum representations of PG and HNG gates, respectively. It can be seen that the quantum cost of these gates equals 4 and 6, respectively. A further reversible gate used in quantum circuits is the Fredkin gate, referred to as a SWAP-controlled gate. If the control input equals 1, the values of the target signals are exchanged [13]. Fig. 1(c) shows the symbol for Fredkin gate and its quantum implementation. As shown, the quantum cost of Fredkin's gate is 5.

3. Proposed integrated structure of forward and reverse converter using reversible logic gates

Due to no transmission of carry digits and the use of parallel calculations, calculations in RNS have a high speed and the limited transmission of the transfer digit is done limitedly within the scaled circuits. Since the forward and reverse converters constitute system overhead and lead to a reduction in the speed achieved due to the parallelization of the moduli arithmetic circuits in the system, a wealth of research has been done for the efficient design of these converters.
In this section, the arithmetic relationships related to forward and reverse converters are discussed separately for the two moduli sets using conventional gates, and in the subsequent section, the integration of the converters for each module set using a reversible logic gate has been described to achieve a composite integrated design.

3.1 Rewriting arithmetic relations of forward and reverse conversion

While designing the forward converter, first, the remainder of the division of the input weighting operand on each of the moduli is obtained; Thus, the forward converter possesses

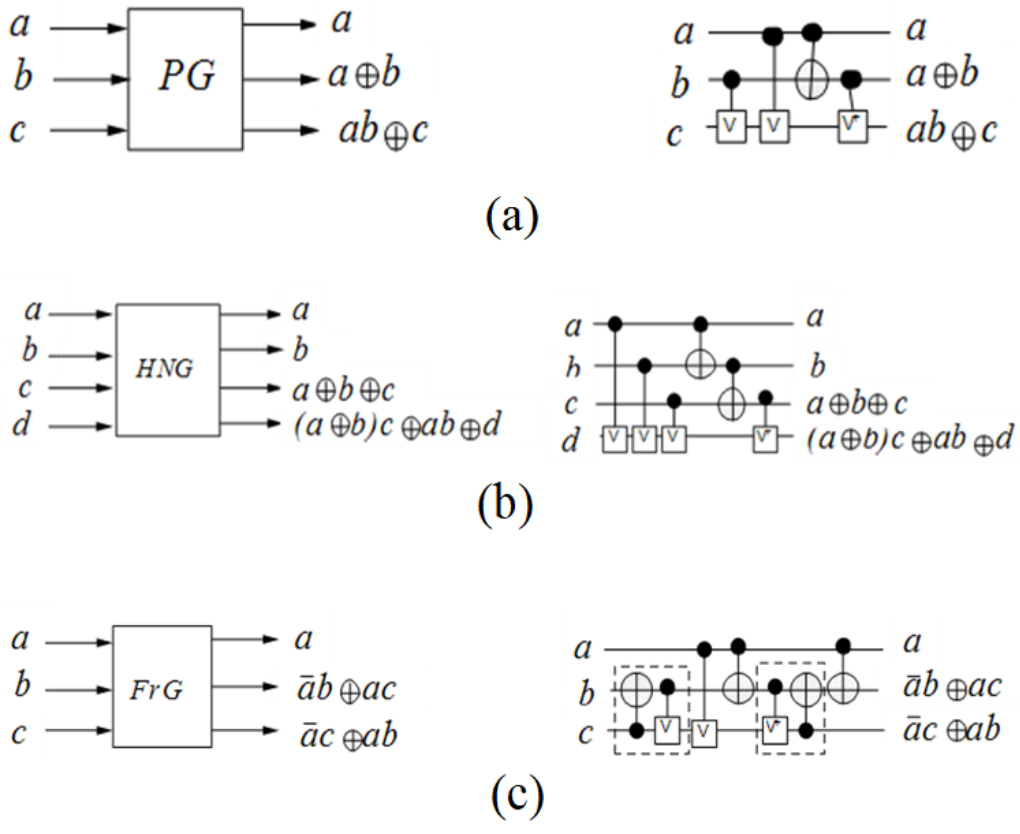

Figure 1. Reversible gates and quantum representation of (a) Peres [12], (b) HNG [10], (c) Fredkin [13].

a parallel and independent structure for each module. To convert the weighted number X based on the moduli set m_1, m_2,..., $m_n$to the residuals (xi's), the relation 4 can be used:

$$x_i = |X|_{m_i} = |X_{MB-1}...X_1X_0|_{m_i} \quad \text{for } i = 1,...,n \qquad (4)$$

where MB is the number of bits of the dynamic range (that is, M), m_i shows the target moduli [1] and X_i demonstrates the i-th *bit* of the binary number X. Based on the number of bits of the m_i moduli expressed by m_{ib}, the operand bits are divided into m_{ib} *bit* categories:

$$x_i = |X|_{m_i} = \left| \begin{matrix} 2^{MB}(X_{2MB-1}...X_{Kmib}) + ...+ \\ 2^{m_{ib}}(X_{2m_{ib-1}}...X_{m_{ib}}) + (X_{m_{ib-1}}...X_0) \end{matrix} \right|_{m_i} \qquad (5)$$

Now there is no need to divide to calculate the remainder and it is obtained by addition and multiplication. In other words, relation (5) can be rewritten as follows:

$$x_i = |\sum_{j=0}^{k} f_i|_{m_i} \qquad (6)$$

where binary vectors f_i are defined as follows:

$$f_i = 2^{jm_{ib}}(X_{(j+1)m_{ib-1}}...X_{jm_{ib}}) \qquad (7)$$

Note that the moduli are often in power of 2 (such as $2^n \pm 1$), and thus the multiplications in (5) are eliminated (changed to shifts). The number and type of moduli of a set directly affect the performance of RNS-based arithmetic systems. Some moduli sets enable straightforward formulations for the reverse conversion and hence efficient converters, while others yield more efficient RNS arithmetic units. By balancing the values of different moduli and using the moduli sets of class c, for example by choosing the set of moduli $\{2^{2n}, 2^n-1, 2^n+1\}$ and $\{2^{2n}+1, 2^n+1, 2^n-1\}$ as moduli set number 1 and 2, respectively, as a case study, the residuals are obtained as follows [24, 25]. Note that the diagrams are easily generalizable. The set of moduli used in this research, due to the large dynamic range of $4n+1$ and $4n$ *bits*, as well as the use of well-formed moduli, yields RNS with fairly high efficiency.

Rewriting the relations of forward and reverse converters in moduli set 1:

$$x_i = |X|_{m_i} = \left| \begin{matrix} 2^{2n}(X_{3n-1}...X_{2n}) + ... + 2^{2n}(X_{2n-1}...X_{2n}) \\ +(X_{n-1}...X_0) \end{matrix} \right|_{m_i} \qquad (8)$$

Considering that $|2^{2n}|_{2^n} = |2^n|_{2^n} = 0$, x_1 is obtained as follows:

$$x_1 = |X|_{2^n} = X_{n-1}...X_0 \qquad (9)$$

On the other hand, considering that $|2^{2n}|_{2^n-1} = |2^n|_{2^n-1} = 1$, x_2 is obtained as follows:

$$x_2 = |X|_{2^n-1} = |\underbrace{X_{3n-1}...X_{2n}}_{f_2} + \underbrace{X_{2n-1}...X_n}_{f_1} + \underbrace{X_{n-1}...X_0}_{f_0}|_{2^n-1} \qquad (10)$$
$$= |f_2 + f_1 + f_0|_{2^n-1}$$

Finally, considering that $|2^n|_{2^n+1} = -1 and |2^{2n}|_{2^n+1} = 1, x_3$ is obtained as follows:

$$x_3 = |X|_{2^n+1} = |\underbrace{X_{3n-1}...X_{2n}}_{f_2} + \underbrace{X_{2n-1}...X_n}_{f_1} + \underbrace{X_{n-1}...X_0}_{f_0}|_{2^n-1}$$
$$= |f_2 - f_1 + f_0|_{2^n-1} \qquad (11)$$

The reverse conversion algorithm based on the set $\{m_1, m_2, \ldots, m_n\}$ and the following arithmetic relations do the reverse conversion. In other words, it converts a residual number such as $\{x_1, x_2, \ldots, x_n\}$ into the ordinary weighted equivalent of X [20].

$$X = v_n \prod_{i=1}^{n-1} m_i + ... + v_3 m_2 m_1 + v_2 m_1 + v_1 \qquad (12)$$

Equation (12) is the basic formula of the permutation algorithm, whose input is the residuals and the values obtained in measurements, and the output is the calculated value of the weighted number. The v_i coefficients are obtained as follows in terms of remainders and multiplicative inverses:

$$v_n = \left| (x_n - v_1|m_1^{-1}|_{m_n} - v_2)|m_2^{-1}|_{m_n} - ... - v_{n-1}|m_{n-1}^{-1}|_{m_n} \right|_{m_n} \qquad (13)$$

Now, the coefficients related to the MRC conversion are rewritten as follows. It should be noted that k_i means the reverse of multiplication:

$$v_2 = \left| \underbrace{k_1 x_2}_{v_{21}} + \underbrace{(-k_1 v_1)}_{v_{22}} \right|_{m_2} = |v_{21} v_{22}|_{m_2} = \left| \sum_{i=1}^{2} v_{2i} \right|_{m_2} \qquad (14)$$

$$v_3 = \left| \underbrace{k_3 k_2 x_3}_{v_{31}} + \underbrace{(-k_3 k_2 v_1)}_{v_{32}} + \underbrace{(-k_3 v_2)}_{v_{33}} \right|_{m_3} = |v_{31} v_{32} v_{33}|_{m_2}$$
$$= \left| \sum_{i=1}^{3} v_{3i} \right|_{m_3} \qquad (15)$$

$$v_n = \left| (x_n - v_1|m_1^{-1}|_{m_n} - v_2)|m_2^{-1}|_{m_n} - ... - v_{n-1}|m_{n-1}^{-1}|_{m_n} \right|_{m_n} \qquad (16)$$

Rewriting the relations of forward and reverse converters in moduli set 2:

$$x_i = |X|_{m_i} = \left| \begin{matrix} 2^{3n}(X_{4n-1}...X_{3n}) + ... + 2^n(X_{2n-1}...X_n) \\ +(X_{n-1}...X_0) \end{matrix} \right|_{m_i} \qquad (17)$$

Considering that $|2^n|_{2^n+1} = -1$ and $|2^{2n}|_{2^n+1} = 1$, x_1 and x_2 are obtained as follows:

$$x_1 = |X|_{2^n+1} = \left| \begin{matrix} X_{4n-1}...X_{2n+1} \times 2^{2n+1} \\ +X_{2n}...X_0 \end{matrix} \right|_{2^{2n}+1} = -2f_1' : f_0' \qquad (18)$$

$$x_2 = |X|_{2^{2n}+1} = \left| \begin{matrix} \underbrace{X_{4n-1}...X_{3n}}_{f_3} \times 2^{3n} + \underbrace{X_{3n-1}...X_{2n}}_{f_2} \times 2^{2n} \\ + \underbrace{X_{2n-1}...X_n}_{-f_1} \times 2^n + \underbrace{X_{n-1}...X_0}_{f_0} \end{matrix} \right|_{2^n+1} \qquad (19)$$
$$= |-f_3 + f_2 - f_1 + f_0|_{2^n+1}$$

On the other hand, considering that $|2^{2n}|_{2^n-1} = |2^n|_{2^n-1} = 1$, x_3 is obtained as follows:

$$x_3 = |X|_{2^n-1} = \left| \underbrace{X_{4n-1}...X_{3n}}_{f_3} \times 2^{3n} + \underbrace{X_{3n-1}...X_{2n}}_{f_2} \times 2^{2n} + \underbrace{X_{2n-1}...X_n}_{f_1} \times 2^n + \underbrace{X_{n-1}...X_0}_{f_0} \right|_{2^n-1} = \left| f_3 + f_2 + f_1 + f_0 \right|_{2^n-1} \quad (20)$$

The algorithm of new CRT based on the set $\{m_1, m_2, \ldots, m_n\}$ is calculated by Equation (12), which serves as the basic formula of the algorithm of the new CRT. In this equation, the input values, the residuals, the number of moduli, and the output are the weight number calculated (in this formula, n is the number of moduli). Thus, it converts a residual number such as $\{x_1, x_2, \ldots, x_n\}$ into the normal weighted equivalent of X [20].

$$X = x_1 + m_1 \left| \begin{matrix} k_1(x_2 - x_1) + m_2 k_2 (x_3 - x_2) + m_2 m_3 k_3 \\ +(x_4 - x_3) + m_2 m_3 ... m_{n-1} k_{n-1} (x_n - x_{n-1}) \end{matrix} \right|_{m_2 ... m_n} \quad (21)$$

The k_i coefficients show the multiplicative reverse of the mi moduli. The multiplicative reverses are calculated separately and inserted into the above equation to calculate the coefficients of the new CRT algorithm.

$$\left| k_{n-1} m_1 ... m_{n-1} \right|_{m_n} = 1 \quad (22)$$

$$\left| k_1 \times m_1 \right|_{m_2 m_3} = \left| k_1 \times (2^{2n} + 1) \right|_{2^{2n}-1} = 1 \Rightarrow k_1 = 2^{2n} - 1 \quad (23)$$

$$\left| k_2 \times m_1 m_2 \right|_{m_3} = \left| k_2 \times (2^{2n} + 1)(2^n + 1) \right|_{2^{2n}-1} = 1 \Rightarrow k_2 = 2^{n-2} \quad (24)$$

Considering that we have the coefficients' values, the value of X is calculated from the Equation (21) in terms of the remainders. Note that by choosing the moduli in the form of 2^n and $2^n \pm 1$, the multiplications are converted into shifts, and hence there is no need for the use of the multiplier in the reverse converter.

4. The suggested method

In both forward and reverse conversion operations, several separate moduli additions are required for each module, which is taken into consideration in the design of hybrid converters.

The control sign enables a switch between forward and reverse converters. If control = 1, the converter works in forward mode (a binary weighted number such as X is inserted, and then the remainders of the weighted number X, i.e. $(x_1, x_2, \ldots, x_n)$ are shown in the output), and if control = 0, the converter functions in reverse mode (some residuals such as $(y_1, y_2, \ldots, y_n)$ is inserted, and then the weighted equivalent of the residuals y_i, i.e. the normal weighted number Y, is eliminated from it). Therefore, by replacing forward and reverse converters with a hybrid converter, the required hardware and control signals are reduced, leading to an increase in the overall system performance (see Fig. 2).

4.1 Rewriting arithmetic relations of forward and reverse conversion based on reversible logic for moduli set 1

Now, based on Equations (12) to (16) for the reverse converter, and (8) to (11) for the forward converter, the following relations can be used to design the forward /reverse conversion unit (Table 1).

$$Y = h_n \prod_{i=1}^{n-1} m_i + ... + h_3 m_2 m_1 + h_2 m_1 + h_1$$

$$\text{where} = \begin{cases} x_i = h_i \quad \text{for } i = 1, ..., n \\ h_i = \begin{cases} \left| \sum_{j-1}^{i} v_{ij} \right|_{m_i} & \text{if control} = 0 \\ \left| \sum_{j-0}^{k} f_i \right|_{m_i} & \text{if control} = 1 \end{cases} \end{cases} \quad (25)$$

The forward converter for the moduli 2^{2n} needs no special computing circuit, and it is enough to select the $2n$ low bits of the input value.

$$x_1 = |X|_{2^{2n}} = |X_{4n}...X_1 X_0|_{2^{2n}} = \underbrace{X_{2n-1}...X_1 X_0}_{2n\ bits} \quad (26)$$

The second and third residuals are calculated based on the moduli $2^n - 1$ and $2^{n+1} - 1$, respectively, as follows:

$$x_2 = \left| \sum_{i=0}^{4} f_i \right|_{2^n-1}, \text{ where} \quad (27)$$

$$f_i = \underbrace{X_{(i+1)n-1}...X_{in+1}X_{in}}_{n\ bits} \quad \text{for } i = 0 \text{ to } 3$$

$$x_3 = \left| \sum_{i=0}^{3} g_i \right|_{2^{n+1}-1}, \text{ where} \quad (28)$$

$$g_i = \underbrace{X_{(i+1)n+1}...X_{in+i+1}X_{in+i}}_{n+1\ bits} \quad \text{for } i = 0 \text{ to } 2$$

To obtain the residuals, some binary vectors can be simply added. The relation (10) can be used to calculate x_2 and x_3. That is, adders with *n-bit* and *n+1-bit* carry digit storages implemented using the HNG reversible gate can be used.

Table 1. Inputs and outputs of the hybrid forward and reverse conversion unit.

Control signal	Working mode	Input	Output
0	Reverse converter	$y_1, y_2, \ldots, y_n$	Y
1	Forward converter	X	$x_1, x_2, \ldots, x_n$

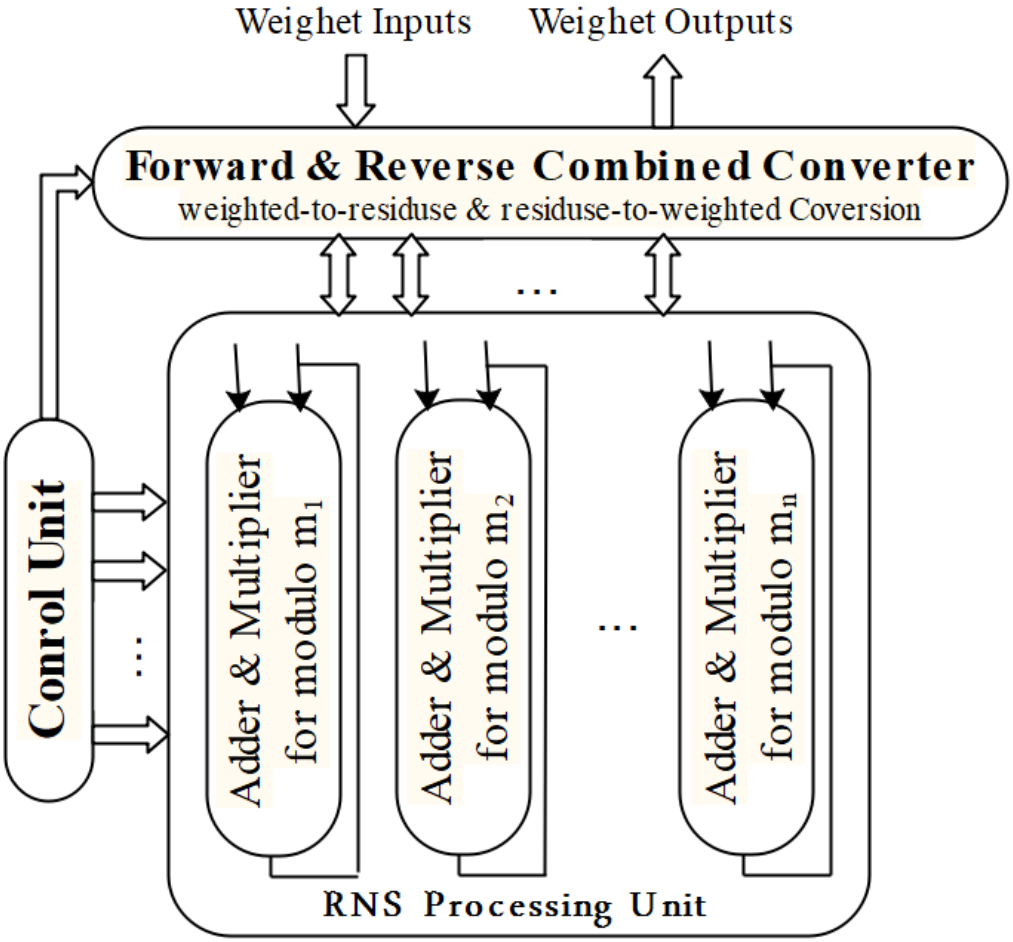

Figure 2. The proposed structure of the residue number system.

This is accompanied by a propagation adder with the ability to rotate the final *n-bit* and *n+1-bit* carry digits, respectively, which is implemented by PG and HNG reversible gates [10] (Fig. 3).

The hardware structure of the reverse converter for the set of ternary modules $\{2^{2n},\ 2^n-1,\ 2^{n+1}-1\}$, based on CRT, has been completely shown in [21]. The following relation is simply used to convert the residuals into weighted equivalents, and after simplifying and using the CSAs and ripple carry propagate adders based on reversible gates, it has been implemented according to the method presented in [10].

$$Y = 2^{2n}(2^n-1)v_3 + 2^{2n}v_2 + v_1$$

$$\text{where} = \begin{cases} v_1 = x_1 \\ v_2 = |w_1 + w_{21} + w_{22}|_{2^n-1} \\ v_3 = |w_3 + w_4 + w_{51} + w_{52}|_{2^{n+1}-1} \end{cases} \tag{29}$$

Where the w_i vectors are as follows:

$$w_1 = \underbrace{x_{2,n-1}\ldots x_{2,1}, x_{2,0}}_{n\ bits} \tag{30}$$

$$w_{21} = \underbrace{\bar{x}_{1,n-1}\ldots\bar{x}_{1,1}\bar{x}_{1,0}}_{n\ bits} \tag{31}$$

$$w_{22} = \underbrace{\bar{x}_{1,2n-1}\ldots\bar{x}_{1,n+1}\bar{x}_{1,n}}_{n\ bits} \tag{32}$$

$$w_3 = \underbrace{v_{2,n-1}\ldots v_{2,1}v_{2,}0}_{n\ bits} \tag{33}$$

$$w_4 = \underbrace{\bar{x}_{3,n-3}\ldots\bar{x}_{3,1}\bar{x}_{3,0}}_{n-2\ bits}\bar{x}_{3,n}\ldots\bar{x}_{3,n-1}\bar{x}_{3,n-2} \tag{34}$$

$$w_{51} = \underbrace{x_{1,n-3}\ldots x_{1,1}x_{1,0}}_{n-2\ bits}x_{1,n}\ldots x_{1,n-1}x_{1,n-2} \tag{35}$$

$$w_{52} = \underbrace{x_{1,2n-2}\ldots x_{1,n+2}x_{1,n+2}}_{n-2\ bits}00x_{1,2n-1} \tag{36}$$

Fig. 4 shows the hardware structure of the reverse converter for the moduli set 1 based on relations (29) to (36). As seen, the CSAs and the ripple carry propagate adders according to the method presented in [10] were used.

5. Rewriting arithmetic relations of forward and reverse conversion based on reversible logic for moduli Set 2

In this section, based on relations (21) to (24) for the reverse converter, and (17) to (20) for the forward converter, and considering that the inverse converter is 2*n bits* and, on the other hand, the forward converter is *n-bit* for one module and *n+1-bit* for the other module, the control unit can use one of the *n-bit* adders for low-value *n bits* and another *n-bit*

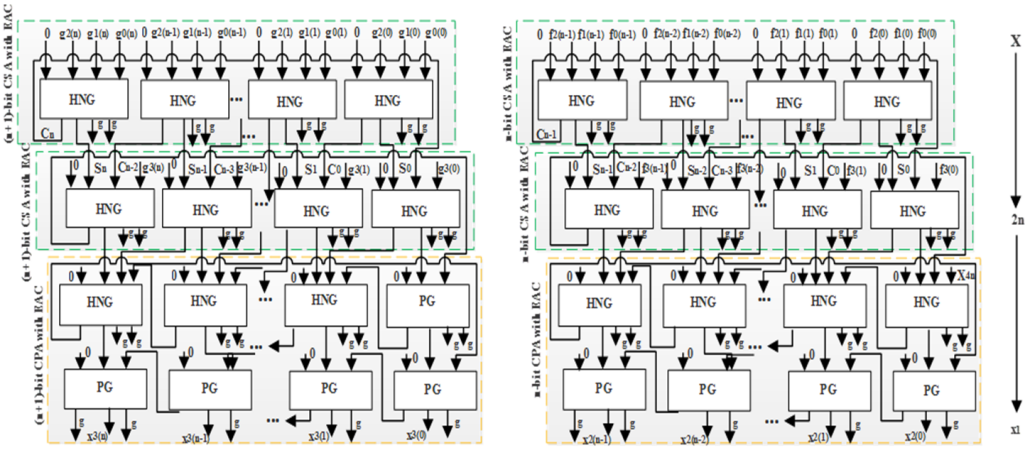

Figure 3. Reversible forward converter for the module set $\{2^{2n},\ 2^n-1,\ 2^{n+1}-1\}$

adder for high-value *n bits*.

To fully demonstrate the characteristics of the proposed method, the design of the forward and reverse converters for this module set is discussed. As seen from relations (17) to (20), to obtain the residuals in the forward converter for the set of modules $\{2^n-1,\ 2^n+1,\ 2^{2n}+1\}$, a number of binary vectors can be simply added using relation (10). That is, adders with *n-bit* and *n+1-bit* carry digit storages implemented using HNG reversible gate can be used. This is accompanied by a propagation adder with the ability to rotate the final *n-bit* and *n+1-bit* carry digits, respectively, which is implemented by PG and HNG reversible gates [11] (Fig. 5 The reverse converter for the module set $\{2^n-1,\ 2^n+1,\ 2^{2n}+1\}$ is obtained in (21) to (26), and then it is simplified as a moduli addition.

In the hardware structure of the reverse converter for the set of three modules $\{2^n-1,\ 2^n+1,\ 2^{2n}+1\}$, by placing the values of the modules, multiplicative reverse, and using the algorithm of the new CRT, it is enough to use the following relations to convert the remainders into a weighted equivalent. Based on the method presented in [10], After simplifying and using CSAs and carry propagate adders with the ability to rotate based on the reversible logic gates, it is implemented.

$$\begin{aligned} X =& x_1+(2^{2n}+1)\\ &\left|2^{2n-1}(x_2-x_1)+2^{n-2}(2^n+1)(x_3-x_2)\right|_{2^{2n}-1} \\ =& x_1+(2^{2n}+1)y \end{aligned} \tag{37}$$

$$\left|y=2^{2n-1}x_2-2^{2n-1}x_1-2^{n-2}(2^n+1)x_2+2^{n-2}(2^n+1)x_3\right|_{2^{2n}-1} \tag{38}$$

$$\begin{aligned} y_1 &= \left|-2^{2n-1}x_1\right|_{2^{2n}-1}\\ &= \left|-2^{2n-1}\left[x_{1(2n)}\times 2^{2n}+x_{1(2n-2)}x_{1(0)}\right]\right|_{2^{2n}-1} \end{aligned} \tag{39}$$

$$\begin{aligned} y_{11} &= \left|-2^{2n-1}\left[x_{1(2n-1)}\ldots x_{1(0)}\right]\right|_{2^{2n}-1}\\ &= \bar{x}_{1(0)}\bar{x}_{1(2n-1)}\ldots\bar{x}_{1(1)} \end{aligned} \tag{40}$$

$$y_{12} = \left|-2^{2n-1}\left[x_{1(2n)}\times 2^{2n}\right]\right|_{2^{2n}-1} = \bar{x}_{1(2n)}\underbrace{1\ldots.1}_{2n-1} \tag{41}$$

$$\begin{aligned} y_2 &= \left|2^{2n-1}x_2-2^{n-2}(2^n+1)x_2\right|_{2^{2n}-1}\\ &= \left|2^{2n-1}x_2-2^{n-2}x_2-2^{n-2}x_2\right|_{2^{2n}-1}\\ &= \left|2^{2n-2}x_2+2^{2n-2}x_2-2^{2n-2}x_2-2^{n-2}x_2\right|_{2^{2n}-1}\\ &= \left|2^{2n-2}x_2-2^{n-2}x_2\right|_{2^{2n}-1} \end{aligned} \tag{42}$$

$$y_{21} = \left|2^{2n-2}x_2\right|_{2^{2n}-1} = x_{2(1)}x_{2(0)}\underbrace{0\ldots0}_{n-1}x_{2(n)}x_{2(2)} \tag{43}$$

$$y_{22} = \left|2^{n-2}x_2\right|_{2^{2n}-1} = 1\bar{x}_{2(n)}\ldots\bar{x}_{2(0)}\underbrace{1\ldots1}_{n-2} \tag{44}$$

$$y'_{12} = 1\underbrace{1\ldots1}_{2n-1}, y'_{22} = \bar{x}_{1(2n)}\bar{x}_{2(n)}\ldots\bar{x}_{2(0)}\underbrace{1\ldots1}_{n-2} \tag{45}$$

$$\begin{aligned} y_3 &= \left|2^{n-2}(2^n+1)x_3\right|_{2^{2n}-1} = \left|2^{n-2}(2^nx_3+x_3)\right|_{2^{2n}-1}\\ &= x_{3(1)}x_{3(1)}\underbrace{x_{3(n-1)}\ldots x_{3(0)}}_{n}\underbrace{x_{3(n-1)}\ldots x_{3(0)}}_{n-2} \end{aligned} \tag{46}$$

$$X = x_1+(2^{2n}+1)y = x_1+\underbrace{2^{2n}y+y}_{s}\quad s=\underbrace{y_{2n-1}\ldots y_0y_{2n-1}\ldots y_0}_{4n} \tag{47}$$

Fig. 6 shows the hardware structure of the reverse converter for module set 2 based on the above arithmetic relations using the CSAs and carry propagate adders with the ability to rotate, according to the method presented in [15].

5.1 Hybrid design of forward and reverse converters using reversible logic gates

Examining the similarity of Fig. 3 and Fig. 4 as well as Fig. 5 and Fig. 6 in modular set 2, and sharing the hardware used by forward and reverse converters, it can be concluded that for the design of forward and reverse converters using reversible PG and HNG gates, CSAs, and the module adders, a single hardware can be obtained for each modular set, separately. So, by using integration relations (48) to (51) for moduli set 1 and integration relations (52) to (54) for moduli set 2, it is possible to design a hardware for forward/reverse hybrid converter.

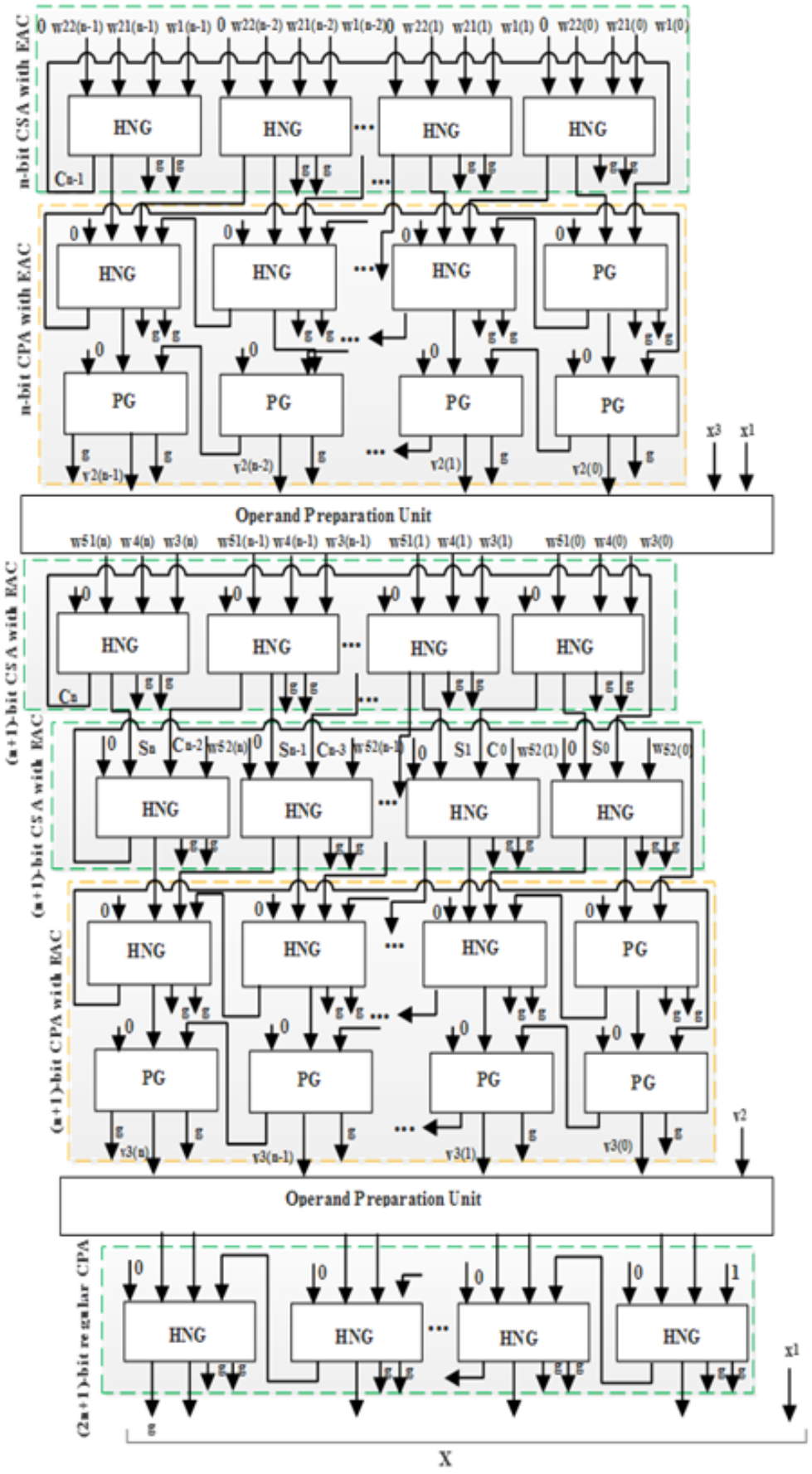

Figure 4. Reversible reverse converter for module set $\{2^{2n},\ 2^{n}-1,\ 2^{n+1}-1\}$.

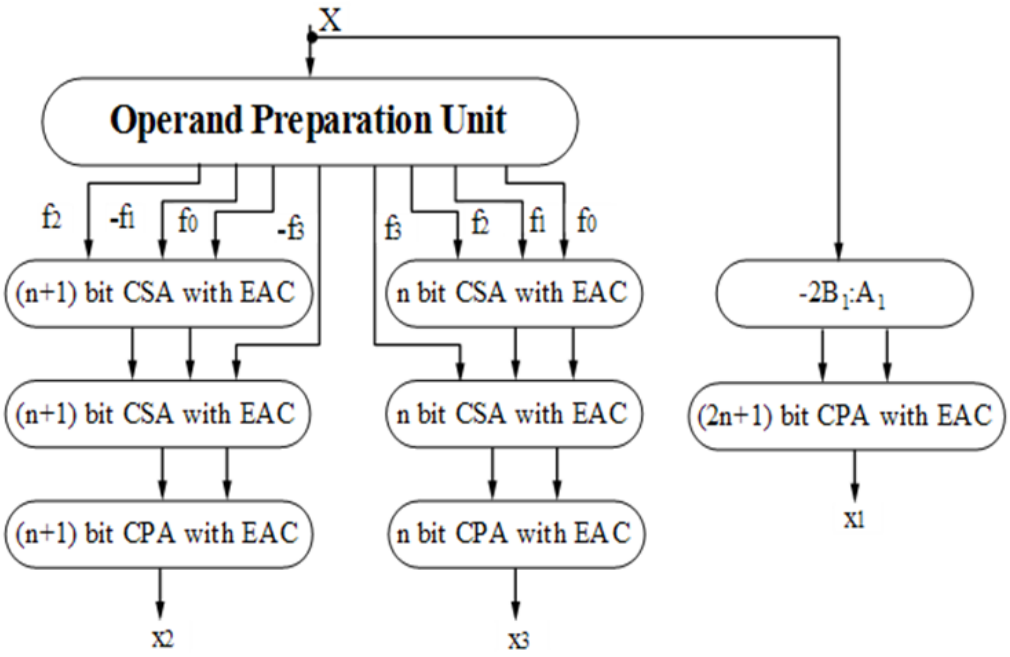

Figure 5. The forward converter for the module set $\{2^n-1,\ 2^n+1,\ 2^{2n}+1\}$.

$$Y = 2^{2n}+1(2^n-1)h_3+2^{2n}h_2+h_1 \tag{48}$$

$$x_1 = h_1 = \underbrace{X_{2n-1}...X_1X_2}_{2n\,bits} \tag{49}$$

$$x_2 = h_2 = \begin{cases} |w_1+w_2+w_{22}|_{2^n-1} & \text{if control} = 0 \\ |f_0+f_1+f_2+f_3+f_4|_{2^n-1} & \text{if control} = 1 \end{cases} \tag{50}$$

$$x_3 = h_3 = \begin{cases} |w_3+w_4+w_{51}+w_{52}|_{2^{n+1}-1} & \text{if control} = 0 \\ |g_0+g_1+g_2+g_3|_{2^{n+1}-1} & \text{if control} = 1 \end{cases} \tag{51}$$

$$x_1 = -2f_1' : f_0' \tag{52}$$

$$x_2 = h_2 = \begin{cases} |-f_3+f_2-f_1+f_0|_{2^n+1} & \text{if control} = 1 \\ |y_{11(0-(n-1))}+y_{21(0-(n-1))}+y'_{22(0-(n-1))}|_{2^n+1} & \\ \text{if control} = 0 \end{cases} \tag{53}$$

$$x_3 = h_3 = \begin{cases} |f_3+f_2+f_1+f_0|_{2^n-1} & \text{if control} = 1 \\ |y_{11(n-(2n-1))}+y_{21(n-(2n-1))}+y'_{22(n-(2n-1))}|_{2^n-1} & \\ \text{if control} = 0 \end{cases} \tag{54}$$

As shown in Fig. 7, by comparing the proposed reversible hybrid converter (Fig. 7) and the ordinary reversible forward and reverse converters (Fig. 3 and Fig. 4), we can use multiplexers to design forward and inverse converters by combined hardware.

Using the control signal, Multiplexers direct the appropriate inputs to the CSAs. If the control signal is zero, the reverse conversion mode is selected, and (y_1, y_2, y_3) will be the inputs of the circuit, and Y will be the output. If the control signal is one, the functional mode of forward conversion is selected, and X will be the input, and (x_1, x_2, x_3) will be the outputs of the circuit.

On the other hand, reversible circuits usually suffer from a large number of inputs and garbage outputs, since converting a digital circuit into a reversible circuit requires adding several garbage outputs and possibly several ancilla inputs.

Considering that increasing the number of circuit inputs and outputs in reversible technologies such as quantum circuits increases the number of qubits and imposes a large cost on the system, reducing the size of reversible circuits by reducing the number of ancilla inputs and garbage outputs in multiplexers would be helpful. In this article, the design of the mux block can be modified to reduce the ancilla inputs and garbage outputs that are imposed due to the overhead caused by the addition of reversible multiplexers to the circuit. Considering that the use of Fredkin gates individually increases the number of blocks that make up the final circuit and naturally increases the ancilla inputs and garbage outputs of reversible circuits, a novel structure designed in this paper can be used to reduce the number of individual blocks.

The more the number of blocks in a circuit, the more ancilla inputs and garbage outputs. As a result, in this proposed hybrid converter, the number of ancilla inputs and garbage outputs decreases as the value of n increases compared to when forward and inverse reversible converters are used separately.

On the other hand, the number and type of adders used in the proposed reversible hybrid circuit in Fig. 8 is the same as [26], that is, in Fig. 7, and only one CSA has been added to cover the fourth operand required for forward conversion. The last reversible carry propagate adder used in Fig. 8 is only needed to perform the reverse conversion. That is, in

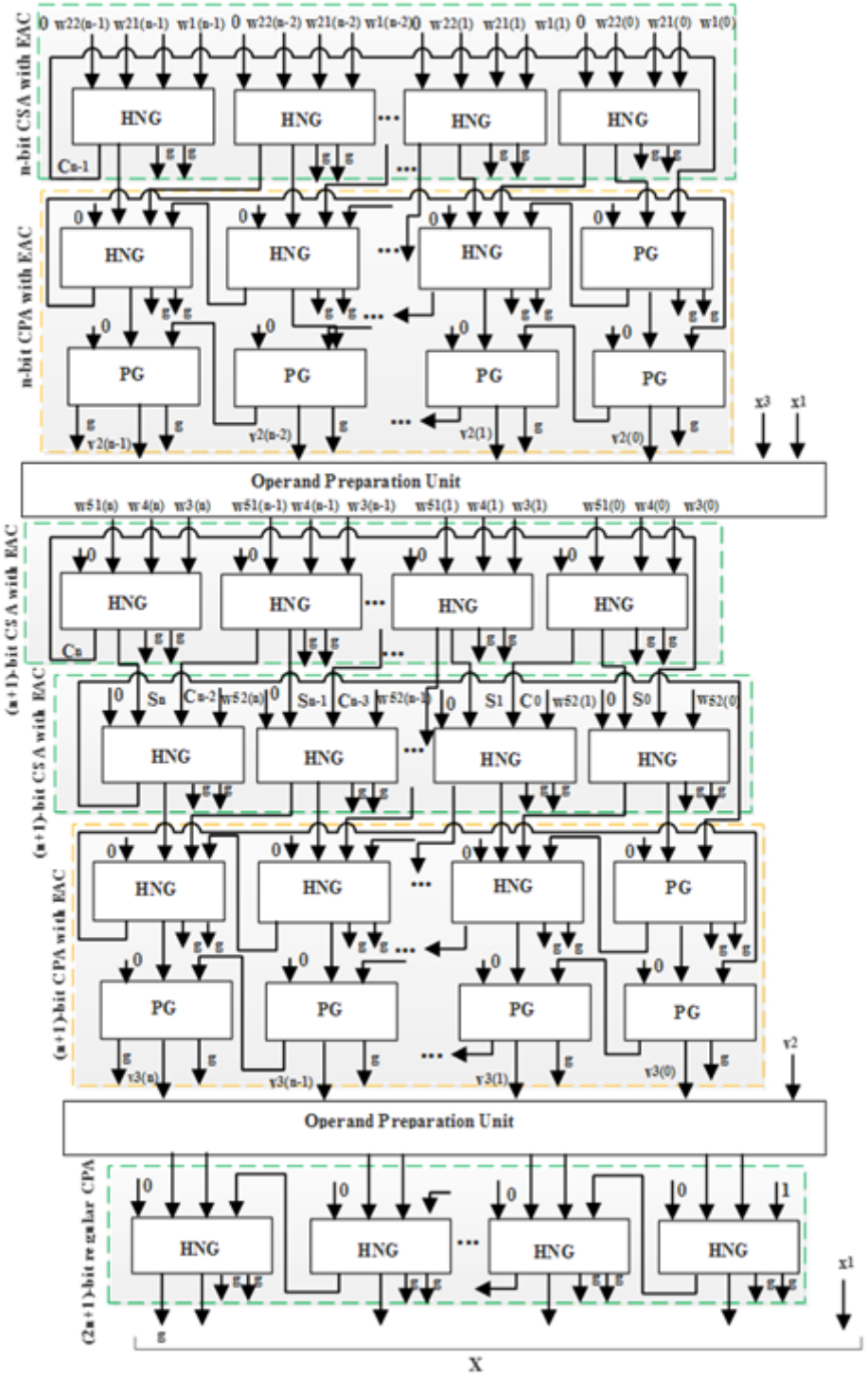

Figure 6. The reverse converter for the module set $\{2^n-1, 2^n+1, 2^{2n}+1\}$.

the forward conversion mode, this adder has no role.

6. Performance evaluation

The most common architectural model for the design of the forward converter is the use of CSAs, followed by a moduli adder [23]. Unlike the forward converter which has a parallel structure, the reverse converter, or the residual bottleneck, is very complex and dependent on the type of conversion algorithm and moduli set. It should be noted that the higher the number of moduli additions and multiplications within the system compared to the number of forward or reverse conversions, the higher the efficiency of the residue number system because the speed advantage resulting from the parallel implementation of additions and multiplications will prevail over the overhead caused by the converter. On the other hand, to achieve a more efficient inverse converter, the following criteria should be considered: 1) choosing a new moduli set, 2) choosing a suitable conversion algorithm, and 3) innovative simplification of the inverse conversion formulas. In choosing between con-

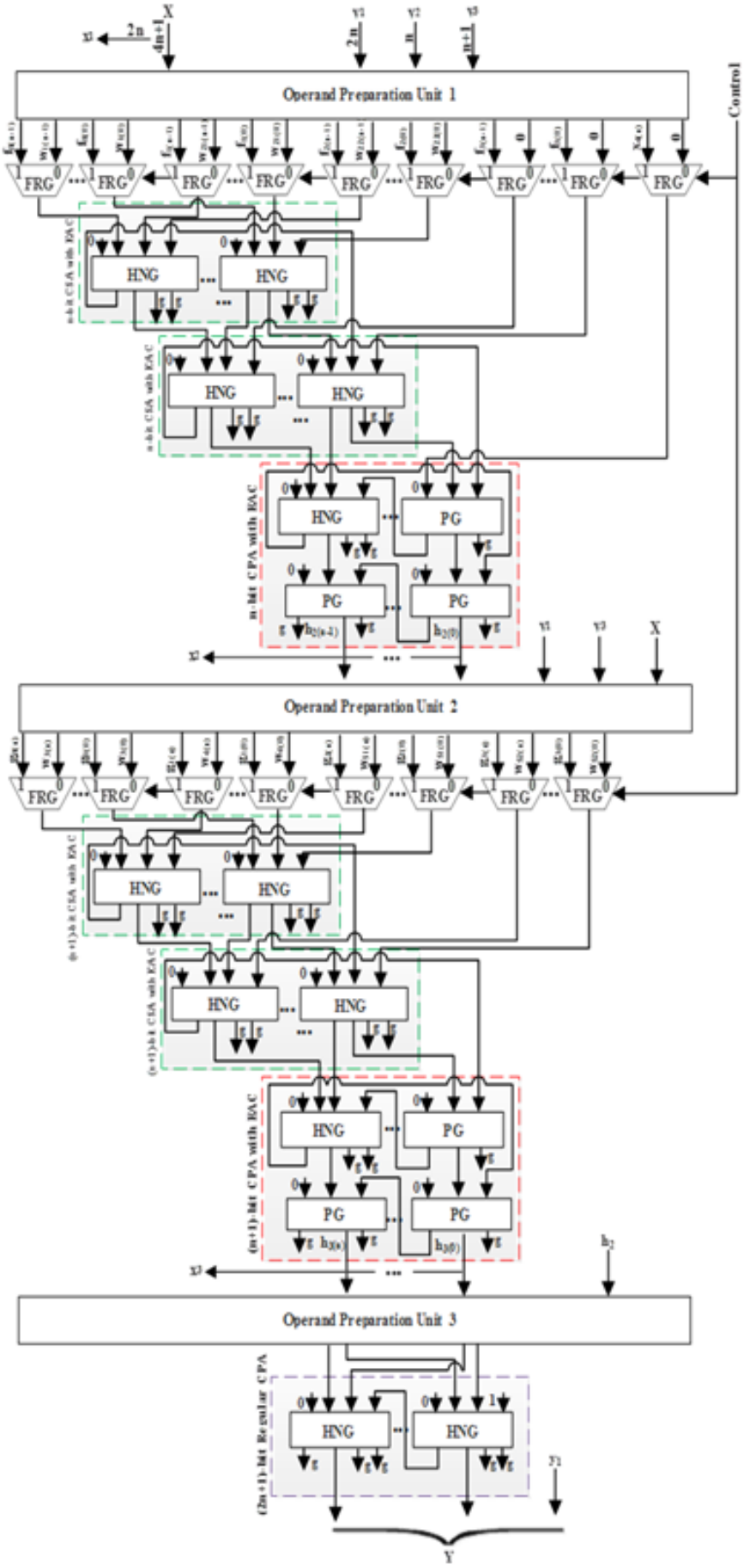

Figure 7. Proposed forward/reverse hybrid converter using reversible logic gates for module set 1.

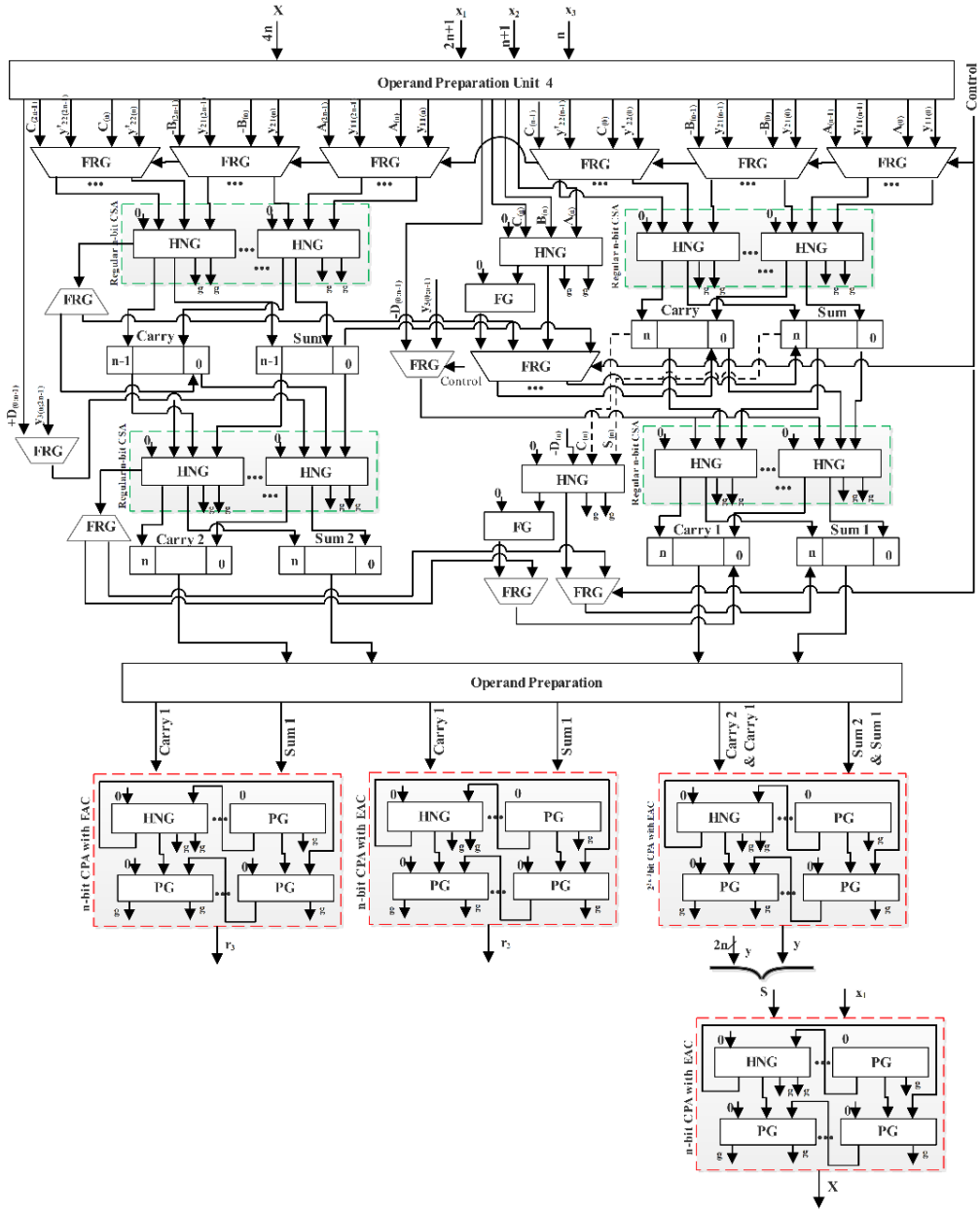

Figure 8. Proposed forward/reverse hybrid converter using reversible logic gates for module set 2

version algorithms, the use of CRTs has received much attention due to their potential for parallel computing, but these theories will increase the system overhead due to the need for a large moduli addition. On the other hand, the mixed-radix conversion using several small-scale moduli additions with an ordinal structure is considered the best conversion algorithm in this research. In addition, the use of special arithmetic hardware components in the inverse converter design according to the needs of the inverse converter will increase the system's performance.

In this study, an independent design for forward and reverse converters in reversible logic was presented. In the suggested method, to design a forward converter, CSA-EAC *k-bit* adders were used that function like an ordinary CSA, where the last carry is re-entered to add the carry digit with the operands and perform a moduli addition. According to [23], as shown in Fig. 1(b), the HNG reversible gate will be converted into an FA by setting the fourth input to the logic zero level. Considering the two-step addition in the binary number system, the required hardware for the ordinary *k-bit* CSA is considered equal to *k* full adder, and its delay is equal to one FA [26].

As can be seen in Fig. 7 and Fig. 8, the proposed reversible forward/reverse hybrid converter for both moduli requires less hardware than when both converters are used simultaneously in the residue number system. This is because all the moduli adders needed in both reversible converters are shared in the hybrid structure. Compared with the ordinary reverse converter [25], in the proposed converter for moduli set 1 and 2, only one and two reversible CSA based on a number of HNG and PG gates and multiplexers based on a number of Fredkin gates have been added in the combined circuit, respectively. In addition, in the working mode of forward conversion, the calculation of

the output residue is done parallel because (27) and (28) are independent, and only some Fredkin gates have been added to the critical path compared to the ordinary forward converter. Therefore, the delay in the combined circuit of the proposed forward/reverse reversible converter has been slightly increased compared to the independent reversible converter because multiplexers (Fredkin gates) and CSAs (HNGs and PGs) have been added to the critical path delay. In general, the proposed design in Fig. 3 and Fig. 4 requires $6n+2$ and $7n+2$ HNG gates and $2n+2$ and $2n+3$ PG gates, respectively. Considering that it is the first attempt to develop a hybrid design of forward and reverse converters in reversible logic, a comparison can be made with independent designs of forward and reverse converters in reversible logic to evaluate either converter's efficiency.

Therefore, the circuits in the forward converter (Fig. 3 and Fig. 5) and reverse converter (Fig. 4 and Fig. 6) in reversible logic are compared with the proposed forward/reverse hybrid converter in reversible logic (Fig. 7 and Fig. 8) based on the moduli set $\{2^{2n},\ 2^n-1,\ 2^{n+1}-1\}$ and $\{2^n-1,\ 2^n+1,\ 2^{2n}+1\}$, respectively, in terms of their efficiency.

As mentioned in the introduction, different criteria are employed to assess reversible circuits. Among the crucial ones are the circuit's gate count (NoG), the quantum cost (QC) which is determined by the number of primary gates needed for circuit implementation, the ancilla input count (NCin) used to balance the number of inputs, and outputs, and the garbage output count (NGout) added to establish a one-to-one correspondence between input and output combinations. These parameters are presented in Tables 2 and 3 for the forward converter, reverse converter, and the proposed converter for moduli sets 1 and 2, respectively. The proposed model focuses on quantum cost reduction. Therefore, as explained in [29], as the quantum cost and the number of gates decrease, the number of ancilla inputs and garbage outputs increases. The benchmark values listed in Tables 2 and 3 were generated by the RCViewer + Analyzer tool. This software, which was introduced and explained in [30], is a viable tool for simulating and analyzing reversible circuits.

As expected, forward and reverse reversible converters, when used independently, have a higher quantum cost and number of gates than the hybrid converter. However, it should be noted that the hybrid converter yields a negligible overhead in terms of the number of ancilla inputs and garbage output compared to using forward/reverse reversible converters. In terms of quantum cost, forward and reverse reversible converters used independently in moduli set 1 ($n=32$) and moduli set 2 yield 19.56% and 19.52% overhead, respectively, compared to when they are used in hybrid mode. Therefore, it can be concluded that the hybrid converter yields less overhead than the independent use of these converters in the reversible logic.

7. Conclusion

In this article, for the first time, a hybrid design of forward and reverse converters based on reversible logic was presented, which was able to reduce the overhead caused by the hardware needed for the reversible forward and reverse converters independently. Therefore, by having such a hybrid converter, the hardware complexity and the total quantum cost of the residue number system in reversible logic are reduced, thus enabling the possibility of its use in applications that require less hardware such as embedded systems. In the future, it is possible to implement hybrid converters as well as other computing and processing circuits that reduce the number of inputs and outputs. This will be achieved by reducing the number of Fredkin gates used as multiplexers in the proposed circuit. Also, other features such as fault tolerance can be added to these circuits, though it should be noted that this feature usually increases the size of the circuit.

Table 2. Performance comparison of converters based on reversible logic (moduli set 1).

Circuit in Reversible Logic	Quantum Cost	Number of Gates	Garbage Outputs	Ancilla Inputs
Forward Converter	1428	260	454	260
Reverse Converter	1624	293	519	293
The Total Forward & Reverse Converters	3052	553	973	553
Proposed Forward/Reverse Converter	2455	454	1031	582
Ref [10]	3037	554	985	551
Ref [27]	3646	661	1163	661

Table 3. Performance comparison of converters based on reversible logic (moduli set 2).

Circuit in reversible logic	Quantum cost	Number of gates	Garbage outputs	Ancilla inputs
Forward Converter	2139	455	712	455
Reverse Converter	2434	449	768	449
The Total Forward & Reverse Converters	4573	904	1480	904
Proposed Forward/Reverse Converter	3680	713	1520	1002
Ref. [11]	3924	736	1273	712
Ref. [28]	3710	719	1204	674

Authors Contributions
All authors have contributed equally to prepare the paper.

Availability of Data and Materials
The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflict of Interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L. S. a. C. C. A. S. Molahosseini. "**embedded systems design with special arithmetic and number systems**.". *Springer International Publishing*, , 2017.

[2] B. Moons, D. Bankman, and M. Verhelst. "**embedded deep neural networks**.". *Embedded Deep Learning: Algorithms, Architectures and Circuits for Always-on Neural Network Processing*, :pp. 1–31, 2019.

[3] L. Sousa, S. Antao, and P. Martins. "**combining residue arithmetic to design efficient cryptographic circuits and systems**.". *IEEE Circuits and Systems Magazine*, 16:pp. 6–32, 2016.

[4] H. Garner. "**the residue number system**.". *IRE Transaction on Electronic Computer*, 8:pp. 140–147, 1959.

[5] M. Mojahed, A. S. Molahossein, and A. A. Emrani Zarandi. "**magnitude comparison and sign detection based on the 4-moduli set** $\{2^n+1,\ 2^n-1,\ 2^n+3,\ 2^n-3\}$.". *Majlesi Journal of Electrical Engineering*, 15:pp. 93–103, 2021.

[6] C. H. Chang, A. S. Molahosseini, A. A. E. Zarandi, and T. F. Tay. "**residue number systems: a new paradigm to datapath optimization for low-power and high-performance digital signal processing applications**.". *IEEE circuits and systems magazine*, 15 (6):pp. 26–44, 2015.

[7] T. M. Conte, E. P. DeBenedictis, P. A. Gargini, and E. Track. "**rebooting computing: the road ahead**. ". *Computer*, 50:pp. 20–29, 2017.

[8] H. A. Mousavi, P. Keshavarzian, and A. S. Molahosseini. "**a novel fast and small xor-base full-adder in quantum-dot cellular automata**. ". *Applied Nanoscience*, :pp. 4037–4048, 2020.

[9] A. Asadpour, A. S. Molahosseini, and A. A. E. Zarandi. "**the use of reversible logic gates in the design of residue number systems**.". *International Journal of Electrical and Computer Engineering (IJECE)*, 13:pp. 2009–2022, 2023.

[10] A. S. Molahosseini, A. Asadpoor, A. A. E. Zarandi, and L. Sousa. "**towards efficient moduliadders based on reversible circuits**.". *in International Symposium on Circuits and Systems (ISCAS), IEEE*, 12: pp. 1–5, 2018.

[11] A. Asadpour, A. S. Molahosseini, and A. A. E. Zarandi. "**the use of reversible logic gates in the design of residue number systems**.". *International Journal of Electrical and Computer Engineering (IJECE)*, 13:pp. 2009–2022, 2023.

[12] A. Peres. "**reversible logic and quantum computers**.". *Physical review A*, 32:pp. 3266, 1985.

[13] E. Fredkin and T. Toffoli. "**quantum mechanical computers**.". *Int. J. Theor. Phys*, 21:pp. 219–253, 1982.

[14] S. M. R. Taha. "**reversible logic synthesis methodologies with application to quantum computing**.". *Springer*, 16, 2016.

[15] E. P. DeBenedictis, J. K. Mee, and M. P. Frank. "**the opportunities and controversies of reversible computing**.". *Computer*, 50:pp. 76–80, 2017.

[16] B. Deng, S. Srikanth, E. Hein, T. M. Conte, E. Debenedictis, J. Cook, and M. P. Frank. "**extending moore's law via computationally error-tolerant computing**.". *ACM Transactions on Architecture and Code Optimization (TACO)*, 15:pp. 1–27, 2018.

[17] H. Sinha and N. Syal. "**design of fault tolerant reversible multiplier**.". *International Journal of Soft Computing and Engineering (IJSCE)*, 1:pp. 120–124, 2012.

[18] N. K. Misra, M. K. Kushwaha, S. Wairya, and A. Kumar. "**cost efficient design of reversible adder circuits for low power applications**.". *arXiv preprint*, : pp. 1509–04618, 2015.

[19] M. B. Ali, H. A. Rahman, and M. M. Rahman. "**design of a high performance reversible multiplier**.". *International Journal of Computer Science Issues (IJCSI)*, 8:pp. 134, 2011.

[20] A. S. Molahosseini, A. A. E. Zarandi, P. Martins, and L. Sousa. "**a multifunctional unit for designing efficient rns-based datapaths**.". *IEEE Access*, 54:pp. 25972–25986, 2017.

[21] M. Mojahed, A. S. Molahosseini, and A. A. E. Zarandi. "**multifunctional unit for reverse conversion and sign detection based on five-moduli set** $\{2^{2\mathbf{n}}, 2^{\mathbf{n}}+1, 2^{\mathbf{n}}-1, 2^{\mathbf{n}}+3, 2^{\mathbf{n}}-3\}$.". *Computer Science*, 22:pp. 101–121, 2021.

[22] P. M. M. Matutino, R. Chaves, and L. Sousa. "**arithmetic-based binary-to-rns converter modulo** $2\mathbf{n} \pm \mathbf{k}$ **jn-bit dynamic range**.". *IEEE Transactions on Very Large Scale Integration (VLSI)*, 23:pp. 603–607, 2015.

[23] M. Haghparast and K. Navi. "**a novel reversible bcd adder for nanotechnology based systems**.". *IAmerican Journal of Applied Sciences*, 5:pp. 282–288, 2008.

[24] A. S. molahosseini, C. Dadkhah, K. Navi, and M. Eshghi. "**efficient mrc-based residue to binary converters for the new moduli sets** $\{22^{\mathbf{n}}, 2^{\mathbf{n}}-1, 2^{\mathbf{n}+1}-1\}$ **and** $\{22^{\mathbf{n}}, 2^{\mathbf{n}}-1, 2^{\mathbf{n}-1}-1\}$.". *IEICE transactions on information and systems*, 9:pp. 1628–1638, 2009.

[25] A. S. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, and S. Timarchi. "**efficient reverse converter designs for the new 4-moduli sets** $\{2^{\mathbf{n}}\text{–}1, 2^{\mathbf{n}}, 2^{\mathbf{n}}+1, 2^{2\mathbf{n}+1}\text{–}1\}$ **and** $\{2^{\mathbf{n}}\text{–}1, 2^{\mathbf{n}}+1, 2^{2\mathbf{n}}, 2^{2\mathbf{n}}+1\}$ **based on new crts**.". *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57:pp. 823–835, 2009.

[26] A. Emrani Zarandi and A. Sabbagh Molahosseini. "**hybrid design of forward and reverse converters: a new approach to reduce hardware complexity of residue number system**.". *TABRIZ JOURNAL OF ELECTRICAL ENGINEERING*, 50:pp. 1315–1328, 2020.

[27] B. K. Raju, P. R. Kumar, and P. B. Rao. "**residue arithmetic's using reversible logic gates**.". *International Conference on Devices, Circuits and Systems (ICDCS),IEEE*, :pp. 1–6, 2014.

[28] S. Shirahatti, R.Shettar, R.Hongal, and U.Malenahalli. "**performance analysis of rns arithmetic operations using reversible logic**.". *International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT),IEEE*, :pp. 1–5, 2022.

[29] M. Mohammadi and M. Eshghi. "**on figures of merit in reversible and quantum logic designs**.". *Quantum Information Processing*, 8:pp. 297–318, 2009.

[30] M. Arabzadeh and M. Saeedi. "**rcviewer +: a viewer/analyzer for reversible and quantum circuits**.". , 2018.