Reverse Converter for the Moduli Set $\{2^n-1, 2^n, 2^n+1\}$ Base on Grouping Number

Saeid Banhanfar[1], Nadali Zarei[2]

[1]**Dept. of engineering, Imam Housein University, Tehran, Iran**

[2]**Dept. of engineering, Imam Housein University, Tehran, Iran**

Abstract

The Residue Number System (RNS) is a non-weighted system that is very efficient in digital signal processing and communicational applications. The previous proposed methods for the residue to binary (R/B) conversions are based on the Chinese Reminder Theorem (CRT) or Mixed Radix Conversion (MRC). These theorems are difficult to implement. In this paper, we present a new high-speed ROM-less residue to binary converter for the three moduli set of $\{2^n-1, 2^n, 2^n+1\}$. Our technique unlike previous methods uses the grouping numbers in dynamic representation range M which its delay is much less than other converters.

Keywords: *residue number system, reverse converter, moduli set $\{2^n-1, 2^n, 2^n+1\}$, group number*.

1. Introduction

Residue Number System is an unconventional system. In this system, an integer X is represented by its reminder modulo a number of different bases. These residue numbers are smaller than the original number in the conventional system, so computations can be done with more speed and low power [1]. The advantages of RNS for implementing digital signal processors for certain applications such as FIR filtering are well-known [2-5]. Some of the more recent applications have been for 1-D filtering [6-8], 2-D filtering [9], video filtering [10], RSA cryptography [11-14], Elliptic curve cryptography [15], m-ary orthogonal keyed communication scheme [16], general purpose RISC DSP [17] and Image processing [18].

The RNS is determined using a set of relative pair wise prime integers positive co-prime integers $\{m_1, m_2, \ldots, m_n\}$ as moduli set. The dynamic range M of that system is given as a product of the moduli m_i where

$$M = \prod_{i=1}^{n} m_i . \qquad (1)$$

Any integer X between 0 and $M-1$ can be uniquely represented as $(x_1, x_2, \ldots, x_n)$. The residues $x_i = |X|_{m_i}$, also called residue digits, are defined as

$$(2)\ x_i = X \bmod m_i , \qquad 0 \le x_i < m_i .$$

The two most important issues for the residue arithmetic are the choice of moduli sets and the conversion of residue to binary numbers. The choice of moduli set in RNS is of continuing interest. Early designs of RNS-based processors were largely based on ROMs which used small set of mutually prime integers to realize a large dynamic range. However, the R/B converters for the general moduli sets are hardware intensive and implemented based on LUTs (Look-up tables). The access time of the LUTs and the need to read these iteratively have made the implementations inefficient for ASIC realization for RNS with large dynamic range. Hence, the more recent trend has been to use moduli sets which can help to eliminate the ROMs. These are known as power-of-two related moduli sets or "conversion-friendly" moduli sets [19].

In residue-based processors, designing an efficient R/B converter is very important. Considerable emphasis has been put on the popular moduli sets like $\{2n-1, 2n, 2n+1\}$, $\{2^n-1, 2^n, 2^n+1\}$, and $\{2^n-1, 2^n, 2^{n-1}-1\}$ [3, 20-25]. The arithmetic processors of these sets and the corresponding residue to binary converters make them very attractive compared with other sets [26]. The powers-of-two related moduli sets have the advantage that all operations required in digital signal processing applications such as modulo addition, modulo subtraction, modulo multiplication, scaling and FIR filtering can be efficiently performed due to the attractive arithmetic properties of these moduli [19].

The three moduli set $\{2^n-1, 2^n, 2^n+1\}$ is of special interest because several operations in this system can be performed efficiently with limited amount or even without ROM. The periodicity properties exhibited by three moduli of this RNS result in superb performance of the binary to residue converter and modulo addition even for large n [27].

Section 2 describes how the conversion of RNS to binary system using the new approach. Section 3 presents the hardware implementation and in section 4, the proposed design is compared with other reported converters.

2. Proposed Method

For residue to binary conversion in moduli set $\{2^n\text{-}1, 2^n, 2^n\text{+}1\}$, we distribute the numbers in dynamic representation range M into several groups and subgroups which a part of this novel idea is presented in [28]. Since, residue representation of X in above moduli set is corresponding with (x_1, x_2, x_3), then the three residues denotes as:

$$\begin{aligned} x_1 &= X \bmod 2^n - 1 = \underbrace{x_{1,n-1}x_{1,n-2}..x_{1,0}}_{n} \\ x_2 &= X \bmod 2^n = \underbrace{x_{2,n-1}x_{2,n-2}..x_{2,0}}_{n} \\ x_3 &= X \bmod 2^n + 1 = \underbrace{x_{3,n}x_{3,n-1}..x_{3,0}}_{n+1} \end{aligned} \quad (3)$$

So, the group number of any residue number in the considered moduli set obtains according to Figure 1.

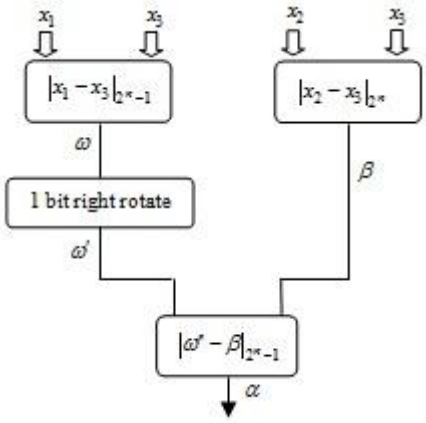

Fig. 1 Group Number Detection.

The number of groups required for this distribution is equal to γ and can be expressed as

$$\gamma = \left\| x_1 - x_3 \right|_{2^n-1} - \left| x_2 - x_3 \right|_{2^n} \Big|_{2^n-1} = 2^n - 1. \quad (4)$$

Therefore, we can concluded that length of any group be called l is given as

$$l = \frac{M}{\gamma} = \frac{(2^n-1).2^n.(2^n+1)}{2^n-1} = 2^n.(2^n+1). \quad (5)$$

In any of these groups there are 2^n subgroups, because

$$\beta = \left| x_2 - x_3 \right|_{2^n}, \quad 0 \le \beta \le 2^n - 1. \quad (6)$$

For example, the value of β for numbers in first group with range $[0, 2^{2n} + 2^n)$ is shown in table1.

Table 1: Distribution of Numbers in subgroups

Number	*Subgroup*
$0 \to 2^n$	0
$2^n\text{+}1 \to 2(2^n\text{+}1) - 1$	1
$2^n\,(2^n\text{+}1) \to 3(2^n\text{+}1) - 1$	2
⋮	
$(2^n\text{–}1)\,(2^n\text{+}1) \to\ 2^n\,(2^n\text{+}1) - 1$	$2^n\text{–}1$

For determination of group number of any residue number, first should be get the value of ω. For clarity, we have exhibited it in range $[0, 2^{2n} + 2^n)$ as follows:

$$\omega = \left| x_1 - x_3 \right|_{2^n-1} \Rightarrow \begin{cases} 0 \le X < 2^n\text{+}1, & \omega = 0 \\ 2^n\text{+}1 \le X < 2\,(2^n\text{+}1), & \omega = 2 \\ 2(2^n\text{+}1) \le X < 3\,(2^n\text{+}1), & \omega = 4 \\ \vdots & \\ (2^{n-1}\text{–}1)(2^n\text{+}1) \le X < 2^{n-1}\,(2^n\text{+}1) & \omega = 2^n\text{–}2 \\ 2^{n-1}(2^n\text{+}1) \le X < (2^{n-1}\text{+}1)(2^n\text{+}1), & \omega = 1 \\ \vdots & \\ (2^n\text{–}2)(2^n\text{+}1) \le X < (2^n\text{–}1)(2^n\text{+}1), & \omega = 2^n\text{–}3 \\ (2^n\text{–}1)(2^n\text{+}1) \le X < 2^n\,(2^n\text{+}1), & \omega = 0. \end{cases} \quad (7)$$

According to (7) and with regard to the product result from moduli subtraction in each group be appeared first, odd values and afterward even respectively. Since, in order to accomplishment of arithmetic operations should be arranged the ω values increasingly, so it is achievable through one bit right rotate. Therefore, if assume $\omega = 0, 2, 4, 6, \ldots, 2^n - 2, 1, 3, .., 2^n - 3$, after 1-bit right rotate, we get $\omega' = 0, 1, 2, \ldots, 2^n - 3, 2^n - 2$.

Thus, by having the values of β and ω', the group number of any residue number in RNS (counting from 0) is defined as

$$\alpha = \left| \omega' - \beta \right|_{2^n-1}, \quad 0 \le \alpha \le 2^n - 2. \quad (8)$$

Table 2 shows the distribution of numbers in dynamic range $[0, 2^{3n} - 2^n)$ which is given as a product of the m_i's in moduli set $\{2^n\text{-}1, 2^n, 2^n\text{+}1\}$.

Table 2: Distribution of Numbers in groups

Number	*Group*
$0 \to 2^n\,(2^n\text{+}1) - 1$	0
$2^n\,(2^n\text{+}1) \to 2[2^n\,(2^n\text{+}1)] - 1$	1
⋮	
$(2^n\text{–}2)[2^n\,(2^n\text{+}1)] \to (2^n\text{–}1)[2^n\,(2^n\text{+}1)] - 1$	$2^n\text{-}2$

Therefore, after the determination of group and subgroup numbers of any number in RNS, its

corresponding number in the binary system is achievable according to equation

$$X = \alpha \cdot 2^n(2^n+1) + \beta.(2^n+1) + x_3 \quad (9)$$

Since, for multiplying one n-bit number in 2 is sufficient to do one bit left cyclic shift of number, then the value of $\alpha.2^n$ is computed as the n-bit left cyclic shift of α. In other words, we have

$$x_{n-1}x_{n-2}x_{n-3}..x_1x_0 \times 2^n = x_{n-1}x_{n-2}x_{n-3}\ldots x_1x_0\underbrace{00\cdots0}_{n} \quad (10)$$

Also, in order to multiplication of n-bit number in 2^n+1, we get

$$x_{n-1}x_{n-2}x_{n-3}..x_1x_0 \times (2^n+1) = \underbrace{x_{n-1}x_{n-2}x_{n-3}\ldots x_1x_0}_{n}\underbrace{x_{n-1}x_{n-2}x_{n-3}\ldots x_1x_0}_{n} \quad (11)$$

Therefore, to achieve the product of a n-bit number by 2^n. (2^n+1) in (9), will be got:

$$x_{n-1}x_{n-2}x_{n-3}..x_1x_0 \times 2^n.(2^n+1) = \underbrace{x_{n-1}x_{n-2}x_{n-3}\ldots x_1x_0}_{n}\underbrace{x_{n-1}x_{n-2}x_{n-3}\ldots x_1x_0}_{n}\underbrace{00\cdots0}_{n} \quad (12)$$

Now equation (9) can be rewritten as

$$X = \underbrace{\alpha\alpha}_{n\ n}\underbrace{00\cdots0}_{n} + \underbrace{\beta\beta}_{n\ n} + \underbrace{x_{3,n}x_{3,n-1}\cdots x_{3,0}}_{n+1} \quad (13)$$

As be seen, X is the 3n bits number. For computation of the lower n bits of X, we can defined θ according to

$$\theta = \underbrace{\underbrace{\beta}_{n} + \underbrace{x_{3,n-1}x_{3,n-2}..x_{3,0}}_{n}}_{n+1} \quad (14)$$

In the other hand, because the lower n bits of X in binary system is equal to the division reminder of number modulo 2^n, then can be concluded in the considered moduli set, no need to computation of θ, due to the lower n bits of θ is achievable through x_2.

$$\theta_{n-1}\theta_{n-2}..\theta_0 = x_2 \quad (15)$$

So, the most significant bit (MSB) of θ is required to computed only and it denotes the carry bit, namely

$$C_{out,\theta} = \theta_n = G_{0:n-1} = C1 \quad (16)$$

As we know, the x_3 is the residue of the binary number mod 2^n+1 and the largest value of x_3 is equal to $1\underbrace{00\cdots0}_{n}$.

In this case, in addition of $\beta + x_3$ which β is a n bit number, we will be $C_{out,\theta} = 0$. Notice that, only while $x_3 = 0\underbrace{XX..X}_{n}$, we can be had $C_{out,\theta} = 1$. Hence, the two bits of $C_{out,\theta}$ and $x_{3,n}$ are not one simultaneously. So we get

$$\delta = C_{out,\theta} + x_{3,n} \quad (17)$$

Therefore, according to the all mentioned issues, can be said

$$\begin{cases} X_{n-1..0} = x_2 = x_{2,n-1}x_{2,n-2}..x_{2,0} \\ C1 = C_{out,\theta} \end{cases}$$

$$\begin{cases} X_{2n-1..n} = \underbrace{\alpha}_{n} + \underbrace{\beta}_{n} + \underbrace{C_{out,\theta}}_{1} + \underbrace{x_{3,n}}_{1} = \alpha + \beta + \delta \\ C2 = C_{out} = MSB(\alpha + \beta + \delta) \end{cases} \quad (18)$$

$$X_{3n-1..2n} = \alpha + C2$$

Due to, the most significant higher 2n bits of X gives as

$$X_{3n-1..n} = \underbrace{\alpha\alpha}_{n\ n} + \underbrace{\beta}_{n} + \underbrace{\delta}_{1} \quad (19)$$

Therefore, the final form of (7) can be rewritten as

$$X = X_{3n-1..n}x_2 \quad (20)$$

Thus, (20) is implemented as a simple concatenation of $X_{n\text{-}1\ldots0}$, $X_{2n\text{-}1\ldots n}$ and also $X_{3n\text{-}1\ldots2n}$. Implementation of equation (20) is shown in Figure 2.

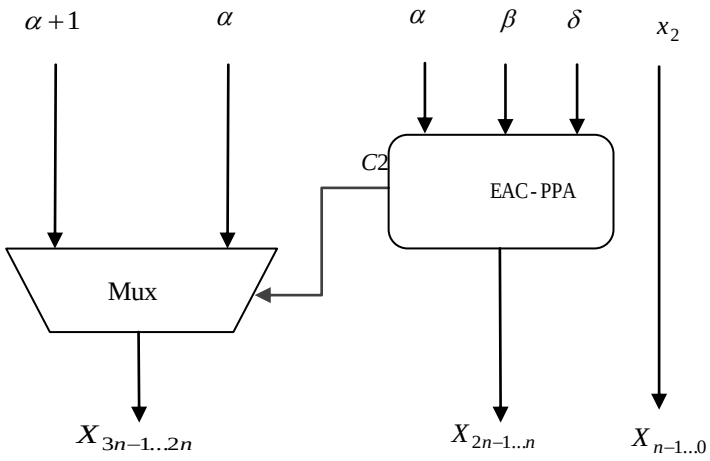

Fig. 2 reverse conversion unit

3. Hardware Structure

The group detection function is determined by Eq.(8) as $\alpha = |\omega' - \beta|_{2^n-1}$. According to [28], since α is computed as a residue modulo 2^n-1 then, instead of subtracting $|\beta|_{2^n-1}$ we can add its additive inverse modulo 2^n-1. An additive inverse modulo 2^n-1 is simply a negation of

binary representation. For simplification reasons the additive inverse of $|\beta|_{2^n-1}$ is denoted as

$$\hat{\beta} = \left| -|\beta|_{2^n-1} \right|_{2^n-1}. \quad (21)$$

So that, the binary form of (21) is $\hat{\beta} = \overline{\beta}_{n-1},...,\overline{\beta}_1, \overline{\beta}_0$. Thus (8) can be rewritten as the sum

$$\alpha = \left| \omega' + \hat{\beta} \right|_{2^n-1}. \quad (22)$$

From [29], an addition modulo (2^n - 1) with redundant zero elimination can be expressed as

$$|a+b|_{2^n-1} = |a+b+c_{out}+p|_{2^n} \quad (23)$$

where c_{out} is a carry bit of $a + b$ addition and $p = 1$ for $a + b = 11\ldots1_2$. The sum $c_{out} + p$ is 0 for $a + b < 2^n - 1$ and 1 for $a + b \geq 2^n - 1$ [30]. By assuming that $C_{in} = c_{out} + p$, the final form of (22) is then

$$\alpha = \left| \omega' + \hat{\beta} + C_{in} \right|_{2^n}. \quad (24)$$

Also, the values of ω and β is given using this way. Notice that, in computing of $\omega = |x_1 - x_3|_{2^n-1}$, because x_3 is a residue number modulo $2^n + 1$ and $x_3 \leq 2^n$ then $|x_3|_{2^n-1}$ is given by OR-ing the least and the most significant bits of x_3. Therefore, binary form of $|x_3|_{2^{n-1}}$ is $\overline{x_{3,0} + x_{3,n}}, \overline{x}_{3,n-1},...,\overline{x}_{3,1}$.

Proposed method for the numbers conversion from residue system to binary system is implemented with parallel prefix structure including parallel-prefix adder and end-around-carry prefix adder, both of which are introduced in [29]. A parallel prefix adder and also parallel prefix adder with end-around-carry are built from elements shown in Figure 3.

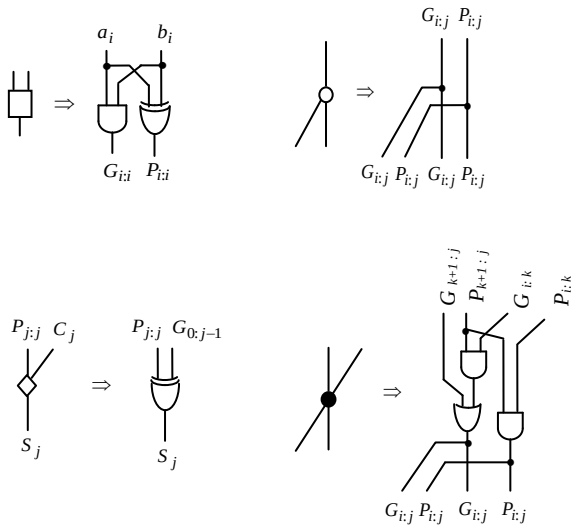

Fig. 3 Blocks of prefix adder structure [30].

The signals $G_{i:j}$ and $P_{i:j}$ are the carry generation and propagation functions from the position i to j. For an addition of two binary vectors $a_{n-1}..a_0$ and $b_{n-1}..b_0$ and for $i < k < j$, these functions can be expressed by logic equations

$$\begin{aligned} G_{i:i} &= a_i \cdot b_i \\ P_{i:i} &= a_i \oplus b_i \\ G_{i:j} &= G_{i:k} \cdot P_{k+1:j} + G_{k+1:j} \\ P_{i:j} &= P_{i:k} \cdot P_{k+1:j}. \end{aligned} \quad (25)$$

The carry signals c_j are equal to $G_{0:j-1}$ and the bits s_j of a final sum are $s_j = P_{j:j} \oplus c_j$. An addition advantage of prefix structures is that the end-around carry can be added in the last stage with a delay cost of two logic levels [30]. The detailed description of this idea is presented in [29].

In this paper, the standard unit-gate model [29] used to estimate the area and time (AT) characteristics of proposed design in order to reverse conversion of numbers by moduli set {2^n-1, 2^n, 2^n+1}. In this model, each two-input monotonic gate (e.g. AND, OR, NAND, NOR), and also XOR/XNOR gate counts as one and two gates respectively.

The Group Number Detection (GND) unit of shown in Fig. 1 comprises three main adders: one modulo (2^n) adder and two adder mod (2^n –1). For calculation of β modulo 2^n, we used the parallel prefix adder from [29] by $A = 5n + (3/2)n\log_2 n + 4$ and $T = 2\log_2 n + 6$. Also, for determination of values ω and α, applied the end-around-carry prefix adder which its hardware and delay are $A = 8n + (3/2)n\log_2 n - 3$ and $T = 2\log_2 n + 6$. Consequently, area and time of GND unit are:

$$\begin{aligned} A_1 &= 21n + \frac{9}{2} n\log_2 n - 2 \\ T_1 &= 4\log_2 n + 12. \end{aligned} \quad (26)$$

As we mentioned already, since lower n bits of the decoded number are available directly as residue corresponding to modulus 2^n, then without need to the hardware components, can be efficiently implemented. $C1$, the product carry bit from $\beta + x_3$ addition be computed using the carry generation unit which is shown in Figure 4.

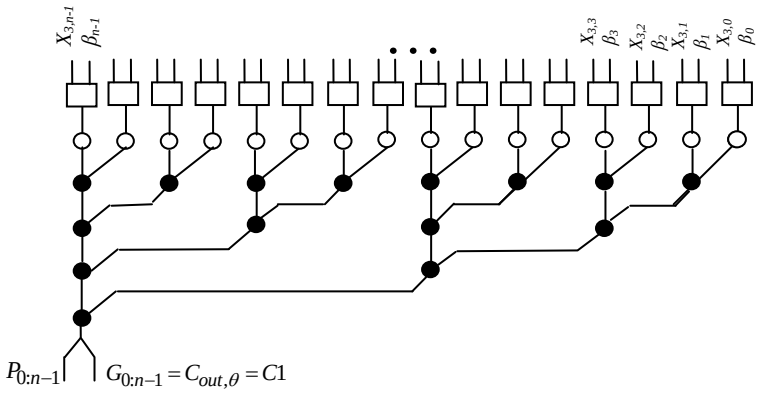

Fig. 4. Carry Generation Unit for n=16

The carry generation unit use (n - 1) black nodes and n input nodes (denoted as square). Its area and time can be expressed as

$$A_2 = 3n + 3(n-1) = 6n - 3$$
$$T_2 = 2\log_2 n + 2 \quad (27)$$

For the calculation of middle n bits from the position X_n to X_{2n-1}, applied the new and modified structure of end-around-carry parallel-prefix adder namely (EAC-PPA) which the output carry signal is determined by $C2$. Figure 5 is used to compute the value of $\alpha + \beta + \delta$ and $C2 = C_{\text{out}}$.

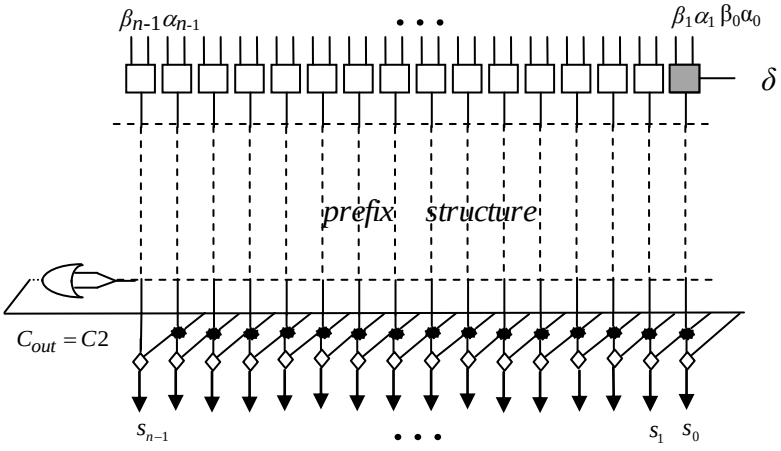

Fig. 5 EAC-PPA

The requirements for the above adder is as follows: n input nods, n output nodes (denoted as lozenge), (n - 1) black node, one additional gate. Notice that first input node is a full adder with area of 4 unit and delay of 2 unit more than half adder. The prefix part of circuit from [29] requires the delay of $2\log_2 n$ logic levels and also, the area of $(3/2)n\log_2 n$. The AT parameters for the shown circuit in Figure 5 are

$$A_3 = 8n + \frac{3}{2} n\log_2 n + 2$$
$$T_3 = 2\log_2 n + 8. \quad (28)$$

So, the required hardware and time for generation of bits n to 2n-1 of the binary number can be expressed as

$$A_4 = 29n + 6n\log_2 n$$
$$T_4 = 6\log_2 n + 20. \quad (29)$$

It is shown in EAC-PPA circuit that delay of generation of $C2$ is equal to $6\log_2 n + 17$.

According to Fig. 2, the most significant n bits of X in binary system is given by α or $\alpha + 1$. The output carry signal from the circuit shown in Fig. 5 be called $C2$ is selecting line of multiplexer which determine whether α be directed to output or $\alpha + 1$. In order to computation of $\alpha + 1$, is sufficient after the value determination of α, be add with 1.

Therefore, the circuit $\alpha + 1$ perform the function of adding 1 to a n-bit input number. Consider $\alpha = \alpha_{n-1}\alpha_{n-2}\ldots\alpha_1\alpha_0, \alpha+1 = \alpha_{n-1}\alpha_{n-2}\ldots\alpha_1\alpha_0 + 1 = e_{n-1}e_{n-2}\ldots e_1e_0$. We have the following equation, which imply that the circuit plus 1 requires n-1 XOR gates and n AND gates plus 1 inverter.

$e_0 = \overline{\alpha_0}$, $e_1 = \alpha_1 \oplus \alpha_1\alpha_0$, $e_i = \alpha_i \oplus \alpha_{i\ldots}\alpha_0$,

$e_{n-1} = \alpha_{n-1} \oplus \alpha_{n-1\ldots}\alpha_1\alpha_0$,

The circuit requires the hardware of 3n and delay of 2 logic levels. Thus, the sum of them consist of

$$A_5 = 23n + \frac{9}{2} n\log_2 n - 2$$
$$T_5 = 4\log_2 n + 14. \quad (30)$$

Since, AT parameters of n bits Mux are 3n and 2 respectively, then area and delay of generation of bits from position 2n to 3n – 1are

$$A_6 = 32n + 6n\log_2 n$$
$$T_6 = 6\log_2 n + 19. \quad (31)$$

Total delay of the circuit is determined by a path consisting one unit of group detection, $C2$ generation unit and multiplexer. The total area and delay of the designed reverse converter circuit are

$$A_{tot} = A_2 + A_3 + A_5 + A_{OR} + A_{Mux} = 41n + 6n\log_2 n - 1$$
$$T_{tot} = \underbrace{T_1 + T_3}_{T_{C2}} + T_{Mux} = 6\log_2 n + 19. \quad (32)$$

4. Conclusions

Reverse converter is one of the most important issues in residue number system. In this paper, a novel and fast algorithm for the conversion of numbers given in RNS $\{2^n\text{-}1, 2^n, 2^n\text{+}1\}$ is presented. Our proposed technique is based on grouping numbers which has significant reduction in delay, compared to other methods. Furthermore, it accomplishes reverse conversion without applying the generic approaches such as CRT and MRC.

References

[1] E. Gholami, R. Farshidi, M. Hosseinzadeh and K. Navi, "High speed residue number system comparison for the moduli set $\{2^n\text{-}1, 2^n, 2^n\text{+}1\}$", Journal of communication and computer, Vol. 6, No. 3, 2009, pp. 40-46.

[2] N. Szabo, R. Tanaka, "Residue arithmetic and its applications to computer technology", New York: MCGraw-Hill, 1967.

[3] M. Soderstrand, M.A.W. Jenkins, G. Jullien and F. Taylor, "Residue number system arithmetic: Modern Applications in Signal Processing", New York: IEEE Press, 1986.

[4] P.V. Ananda Mohan, "Residue number system: algorithms and Architectures", New York: kluwer Academic Publishers, 2002.

[5] F. Taylor, "Residue arithmetic: a tutorial with examples", IEEE Comput Mag, 1984, pp. 50-62.

[6] W. Freking, K. Parhi, "Low-power FIR digital filters using residue arithmetic", In: Conference of the 31st Asilomar conference on signals, systems and computers, 1997.

[7] A. D'Amora, et al, "Residue power dissipation in complex digital filters by using the quadratic residue number system", In: Conference of the 34th Asilomar conference on signals, systems and computers, Vol. 2, 2000, pp.

[8] G. Cardarill, et al, "Low-power implementation of polyphase filters in Quadratic Residue Number System", In: Proceedings of the IEEE international symposium on circuits and systems, vol. 2, 2004, pp. 725-8,

[9] N. Shanbag, R. Siferd, "A single-chip pipelined 2-D FIR filter using residue Arithmetic. IEEE J Solid-State Circuits, 1991, pp. 796-805.

[10] T. Toivonen, J. Heikkila, "Video filtering with format number theoretic transforms using residue number system", IEEE Trans Circuits Syst Video Technol, 2006, pp. 128-38.

[11] J. Schwemmlein, K. Posch , P. Reinhard, "RNS modulo reduction upon a restricted base value set and its applicability to RSA cryptography", 1978, pp. 637-50.

[12] H. Nozaki, M. Motoyama, A. Shimbo and S. Kawamura, "Implementation of RSA algorithms based on RNS Montgomery multiplications", Springer, 2001, pp. 364-76.

[13] J-C. Bajard, L. Kornerup, "An RNS Montgomery modular multiplication algorithm", IEEE Trans Comput, 1998, pp. 769-74.

[14] J-C. Bajard, L. Imbert, "a full RNS Implementation RSA", IEEE Trans Comput 2004, pp. 769-74.

[15] D. Schinianakis, A. Kakarountas, T. Stouraitis, "A new approach to elliptic curve cryptography: an RNS architecture", IEEE Mediterranean electrotechnical conference, 2006, pp. 1241-5.

[16] L-L. Yang, L. Hanzo, "A residue number system based parallel communications scheme using orthogonal signaling", IEEE Trans Veh Technol, 2002, pp. 1534-46.

[17] R. Chaves, L. Sousa, "RDSP: A RISC DSP based on residue number system", In: Proceeding of the Euromicro symposium on digital systems design: architectures, methods and tools, 2003, pp. 128-35.

[18] W. Wei, et al, "RNS applications for digital image processing", In: Proceeding of the 4th IEEE International workshop on system-on-chip for real time application, 2004, pp. 77-80.

[19] P.V. Ananda Mohan, "New reverse converters for the moduli set $\{2^n\text{–}3, 2^n\text{–}1, 2^n\text{+}1, 2^n\text{+}3\}$", International journal of Electronics and Communications, 2007, pp. 643-58.

[20] A. Premkumar, "An RNS to binary converter in a three moduli set with common factors", IEEE Trans. Circuits Syatems-Part II. 1995, pp. 298-301.

[21] A. hiasat, H. Zohdi, "Residue to binary arithmetic converter for the moduli $\{2^k, 2^k\text{–}1, 2^{k-1}\text{–}1\}$", IEEE Trans. Circuits Syatems-Part II. 1998, pp. 204-209.

[22] A. Sweidan, A. Hiasat, "New efficient memoryless, residue to binary converter", IEEE Trans. Circuits systems, 1988, pp. 1441-44.

[23] B.Bernardson, "Fast memoryless, over 64 bits, residue to decimal converter", IEEE Trans. Circuits Systems, 1985, pp. 298-300.

[24] R. Conway, J. Nelson, "Fast converter for 3 moduli RNS using new property of CRT", IEEE Trans Comput. 1999, pp. 852-60.

[25] W. Wang, M. Swamy, M. Ahmad and Y. Wang, "A high speed residue to binary converter for the three-moduli $\{2^k, 2^k\text{–}1, 2^{k-1}\text{–}1\}$" IEEE Trans. Circuits Systems-Part II, 2000, pp.1576-1581.

[26] A. A.Hiasat, "An arithmetic residue to binary conversion technique", Integration, the VLSI Journal, 2003, pp. 13-25.

[27] S. J. Piestrak, "A high-Speed Realization of a Residue to Binary Number System Converter", IEEE Trans on circuits and systems, Vol. 42, No. 10, 1995.

[28] M. Rouhifar, M. Hosseinzadeh, S. Bahanfar and M. Teshnehlab, "Fast Overflow Detection in moduli set $\{2^n\text{-}1, 2^n, 2^n\text{+}1\}$", International Journal of Computer Science Issues, Vol. 8, Issue. 3, 2011, pp. 407-414.

[29] R. Zimmerman, "Efficient VLSI implementation of modulo $(2^n\pm1)$ addition and multiplication", in Proc. 14th IEEE Symp. Comput. Arithm. , 1999, pp. 158-67.

[30] T. Tomczak, "Fast Sign Detection for RNS $\{2^n\text{-}1, 2^n, 2^n\text{+}1\}$", IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 55, Iss. 6, 2008, pp. 1502-11.