ICIC Express Letters
Part B: Applications
Volume **7**, Number **5**, May **2016**

ICIC International ©2016 ISSN 2185-2766
pp. 1111–1117

NEW REVERSE CONVERTER DESIGN FOR 4-MODULI SET $\left\{2^{2n}, 2^{n+1}-1, 2^{n/2}+1, 2^{n/2}-1\right\}$

Siang-Min Siao[1], Ming-Hwa Sheu[1], Yin-Tsung Hwang[2]
Yi-Ci Wang[1] and Kuan-Ta Lee[1]

[1]Department of Electronic Engineering
National Yunlin University of Science and Technology
No. 123, University Road, Section 3, Douliou, Yunlin 64002, Taiwan
{ g9813707; sheumh }@yuntech.edu.tw

[2]Department of Electrical Engineering
National Chung Hsing University
No. 145, Xingda Rd., South Dist., Taichung City 402, Taiwan

Received October 2015; accepted January 2016

Abstract. *This paper presents an efficient reverse conversion algorithm for four-moduli set* $\{2^{2n}, 2^{n+1}-1, 2^{n/2}+1, 2^{n/2}-1\}$ *with* $4n$*-bit dynamic range (DR). Based on new Chinese remainder theorem II (CRT-II), the new algorithm for reverse converter is developed, and then the converter architecture is proposed. The advantage of the proposed design has shorter bit-width CPA (carry propagate adder) with EAC which is the main function in converter hardware. Therefore, it not only improves conversion delay, but also achieves low hardware cost and power consumption. For VLSI implementing based on TSMC 90 nm CMOS process, the proposed converter obtains 32.1% area-power-delay (APD) and 21.8% area-delay (AD) saving at least when compared with related four-moduli sets.*

Keywords: New CRT-II, Dynamic range, Reverse converter, Residue number system

1. **Introduction.** Many modern DSP applications, such as filter [1], error detection [2], cryptography [3], and sign detection [4], etc., need high-performance operations. When compared with the traditional binary number system, the residue number system (RNS) contains non-weighed and carry-free features. These two benefits can reduce power consumption and improve parallelism [5]. In RNS hardware structure, reverse converter is the most complex part, and its circuit performance is mainly dependent on moduli set selection. First, the moduli set can decide system parallelism and complexity of converter. Second, the moduli set needs enough DR magnitude which can support data range of applications. According to the above analysis, three-moduli sets $\{2^n-1, 2^n, 2^n+1\}$ [6] and $\{2^{2n+1}-1, 2^n, 2^n-1\}$ [7] have the disadvantages of low parallelism and less DR. Next, five-moduli sets, such as $\left\{2^n, 2^{2n-1}-1, 2^{n/2}-1, 2^{n/2}+1, 2^n+1\right\}$ [8], and $\left\{2^n, 2^{2n+1}-1, 2^{n/2}-1, 2^{n/2}+1, 2^n+1\right\}$ [9], have higher parallelism. Their converter architectures are more complex so that they have the worst performance in terms of cost and speed. In general, four-moduli sets $\left\{2^{2n}, 2^{n+1}-1, 2^{n/2}+1, 2^{n/2}-1\right\}$ [10], $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$, $\{2^n-1, 2^{2n}, 2^n+1, 2^{2n+1}-1\}$ [11], and $\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$ [12] can provide better condition to design reverse converter to overcome drawbacks of converters of three- and five-moduli sets. From the hardware viewpoint, the converter architecture always contains long bit-width CPA blocks which will dominate and delay. For improving converter performance, reducing CPA blocks and shortening bit-width CPA are essential to design an efficient reverse converter.

This paper presents a new reverse conversion algorithm for four-moduli set $\{2^{2n}, 2^{n+1}-1, 2^{n/2}+1, 2^{n/2}-1\}$. Based on our proposed algorithm, the bit-width length of converter can be reduced. After synthesis based on TSMC 90nm CMOS process, a chip layout of the proposed converter with 64-bit width is accomplished. When compared with recent works, our converter has more than 32.1% APD, and 21.8% AD saving.

The rest of the paper is as follows. Based on new CRT-II, a new reverse algorithm is derived in Section 2 and then present a hardware architecture in Section 3. Section 4 demonstrates all synthesis results in terms of power, delay and area. Under a fair comparison, it will obtain saving percentage of our converter. Finally, Section 5 will make conclusion.

2. **Reverse Conversion Algorithm.** For a co-prime 4-moduli set $\{P_1, P_2, P_3, P_4\} = \{2^{2n}, 2^{n+1}-1, 2^{n/2}+1, 2^{n/2}-1\}$, the residue number $\{x_1, x_2, x_3, x_4\}$ can uniquely represent the binary number X after modulo operation $X \bmod m_i$. Based on new CRT-II, the reverse conversion from the residue number to binary number X can be expressed by

$$X = Z + P_1 P_2 \langle k_1 (Y - Z) \rangle_{P_3 P_4} \tag{1}$$

$$Z = x_1 + P_1 \langle k_2 (x_2 - x_1) \rangle_{P_2} \tag{2}$$

$$Y = x_3 + P_3 \langle k_3 (x_4 - x_3) \rangle_{P_4} \tag{3}$$

where

$$\langle k_1 P_1 P_2 \rangle_{P_3 P_4} = 1, \ \langle k_2 P_1 \rangle_{P_2} = 2^2, \text{ and } \langle k_3 P_3 \rangle_{P_4} = 2^{n/2-1}.$$

The k_1, k_2 and k_3 are the multiplicative inverses.

Then, the multiplicative inverse values of k_1, k_2, k_3 are introduced to system (1). We have

$$X = Z + 2^{2n} \left(2^{n+1} - 1\right) \langle (Y - Z) \rangle_{2^n - 1} \tag{4}$$

where

$$Z = x_1 + 2^{2n} \left\langle 2^2 (x_2 - x_1) \right\rangle_{2^{n+1}-1}, \tag{5}$$

and

$$Y = x_3 + \left(2^{n/2} + 1\right) \left\langle 2^{n/2-1} (x_4 - x_3) \right\rangle_{2^{n/2}-1}. \tag{6}$$

Next, system (5) can be rewritten as

$$Z = x_1 + 2^{2n} H = x_1 + 2^{2n} \langle v_1 + v_2 \rangle_{2^{n+1}-1} = x_1 + 2^{2n} \langle v_1 + v_{21} + v_{22} \rangle_{2^{n+1}-1} \tag{7}$$

where

$$v_1 = \left\langle 2^2 x_2 \right\rangle_{2^{n+1}-1} = \left\langle 2^2 (\underbrace{x_{2,n} \ldots x_{2,0}}_{n+1 \text{ bits}}) \right\rangle_{2^{n+1}-1} = \underbrace{x_{2,n-2} \ldots x_{2,0}}_{n-1 \text{ bits}} \underbrace{x_{2,n} x_{2,n-1}}_{2 \text{ bits}} \tag{8}$$

and

$$\begin{aligned} v_2 = \left\langle -2^2 x_1 \right\rangle_{2^{n+1}-1} &= \left\langle -2^2 (\underbrace{x_{1,2n-1} \ldots x_{1,n+1}}_{n-1 \text{ bits}} \times 2^{n+1} + \underbrace{x_{1,n} \ldots x_{1,0}}_{n+1 \text{ bits}}) \right\rangle_{2^{n+1}-1} \\ &= \left\langle -2^2 (\underbrace{00}_{2 \text{ bits}} \underbrace{x_{1,2n-1} \ldots x_{1,n+1}}_{n-1 \text{ bits}} \times 2^{n+1} + \underbrace{x_{1,n} \ldots x_{1,0}}_{n+1 \text{ bits}}) \right\rangle_{2^{n+1}-1} \\ &= \left\langle \underbrace{\overline{x_{1,2n-1}} \ldots \overline{x_{1,n+1}}}_{n-1 \text{ bits}} \underbrace{11}_{2 \text{ bits}} + \underbrace{\overline{x_{1,n-2}} \ldots \overline{x_{1,0}}}_{n-1 \text{ bits}} \underbrace{\overline{x_{1,n} x_{1,n-1}}}_{2 \text{ bits}} \right\rangle_{2^{n+1}-1} = \langle v_{21} + v_{22} \rangle_{2^{n+1}-1}. \end{aligned} \tag{9}$$

System (6) can be rewritten by using the similar approach

$$Y = x_3 + \left(2^{n/2} + 1\right) K = x_1 + \left(2^{n/2} + 1\right) \langle v_3 + v_4 \rangle_{2^{n+1}-1} \tag{10}$$

where

$$v_3 = \left\langle 2^{n/2-1}x_4 \right\rangle_{2^{n/2}-1} = \left\langle 2^{n/2-1}(\underbrace{x_{4,n/2-1}\ldots x_{4,0}}_{n/2 \text{ bits}}) \right\rangle_{2^{n/2}-1} = \underbrace{x_{4,0}}_{1 \text{ bit}}\underbrace{x_{4,n/2-1}\ldots x_{4,1}}_{n/2-1 \text{ bits}}, \quad (11)$$

and

$$\begin{aligned} v_4 &= \left\langle -2^{n/2-1}x_3 \right\rangle_{2^{n/2}-1} = \left\langle -2^{n/2-1}(\underbrace{x_{3,n/2}}_{1 \text{ bit}} \times 2^{n/2} + \underbrace{x_{3,n/2-1}\ldots x_{3,0}}_{n/2 \text{ bits}}) \right\rangle_{2^{n/2}-1} \\ &= \left\langle -2^{n/2-1}(\underbrace{0\ldots0}_{n/2-1 \text{ bits}}\ \underbrace{x_{3,n/2}}_{1 \text{ bit}} + \underbrace{x_{3,n/2-1}\ldots x_{3,0}}_{n/2 \text{ bits}}) \right\rangle_{2^{n/2}-1} \\ &= \left\langle -(\underbrace{x_{3,n/2}}_{1 \text{ bit}}\ \underbrace{0\ldots0}_{n/2-1 \text{ bits}} + \underbrace{x_{3,0}}_{1 \text{ bit}}\underbrace{x_{3,n/2-1}\ldots x_{3,1}}_{n/2-1 \text{ bits}}) \right\rangle_{2^{n/2}-1}. \end{aligned} \quad (12)$$

Further, the MSB of x_3 has two cases which are $x_{3,n/2} = 0$ and 1. Therefore, v_4 is rewritten as

$$v_4 = \begin{cases} \overline{x_{3,0}x_{3,n/2-1}}\ldots\overline{x_{3,1}}, & \text{if } x_{3,n/2} = 0 \\ 01\ldots1, & \text{if } x_{3,n/2} = 1 \end{cases} \quad (13)$$

After the above bit-level simplification, system (4) can be expressed as

$$X = Z + 2^{2n}\left(2^{n+1}-1\right)T \quad (14)$$

where

$$T = \langle Y - Z \rangle_{2^n-1} = \left\langle x_3 + (2^{n/2}+1)K - x_1 - 2^{2n}H \right\rangle_{2^n-1} = \langle v_5 + v_6 + v_7 + v_8 \rangle_{2^n-1}, \quad (15)$$

$$v_5 = \langle x_3 \rangle_{2^n-1} = \left\langle (\underbrace{x_{3,n/2}\ldots x_{3,0}}_{n/2+1 \text{ bits}}) \right\rangle_{2^n-1} = \underbrace{0\ldots0}_{n/2-1 \text{ bits}}\ \underbrace{x_{3,n/2}\ldots x_{3,0}}_{n/2+1 \text{ bits}}, \quad (16)$$

$$\begin{aligned} v_6 &= \left\langle (2^{n/2}+1)K \right\rangle_{2^n-1} = \left\langle (\underbrace{k_{n/2-1}\ldots k_0}_{n/2 \text{ bits}}\underbrace{k_{n/2-1}\ldots k_0}_{n/2 \text{ bits}}) \right\rangle_{2^n-1} \\ &= \underbrace{k_{n/2-1}\ldots k_0}_{n/2 \text{ bits}}\underbrace{k_{n/2-1}\ldots k_0}_{n/2 \text{ bits}}, \end{aligned} \quad (17)$$

$$\begin{aligned} v_7 &= -\left\langle \underbrace{x_{1,2n-1}\ldots x_{1,n}}_{n \text{ bits}} \times 2^n + \underbrace{x_{1,n-1}\ldots x_{1,0}}_{n \text{ bits}} \right\rangle_{2^n-1} \\ &= \left\langle \underbrace{\overline{x_{1,2n-1}}\ldots\overline{x_{1,n}}}_{n \text{ bits}} + \underbrace{\overline{x_{1,n-1}}\ldots\overline{x_{1,0}}}_{n \text{ bits}} \right\rangle_{2^n-1} = \langle v_{71} + v_{72} \rangle_{2^n-1} \end{aligned} \quad (18)$$

and

$$\begin{aligned} v_8 &= \left\langle -2^{2n}H \right\rangle_{2^n-1} = \left\langle -2^{2n}(\underbrace{0\ldots0}_{n-1 \text{ bits}}\ \underbrace{h_n}_{1 \text{ bit}} \times 2^n + \underbrace{h_{n-1}\ldots h_0}_{n \text{ bits}}) \right\rangle_{2^n-1} \\ &= \left\langle \underbrace{1\ldots1}_{n-1 \text{ bits}}\ \underbrace{\overline{h_n}}_{1 \text{ bit}} + \underbrace{\overline{h_{n-1}}\ldots\overline{h_0}}_{n \text{ bits}} \right\rangle_{2^n-1} = \langle v_{82} + v_{81} \rangle_{2^n-1}. \end{aligned} \quad (19)$$

Finally, the conversion Equation (14) can be expressed as follows:

$$\begin{aligned} X &= Z + 2^{2n}\left(2^{n+1}-1\right)T = x_1 + 2^{2n}H + 2^{2n}\left(2^{n+1}-1\right)T \\ &= x_1 + 2^{2n}\left(H + 2^{n+1}T - T\right) = x_1 + 2^{2n}S \end{aligned} \tag{20}$$

Example 2.1. *Consider the four-moduli set* $\{2^{2n}, 2^{n+1}-1, 2^{n/2}+1, 2^{n/2}-1\} = \{4096, 127, 9, 7\}$*, where* $n = 6$*. The corresponding residue number* $(x_1, x_2, x_3, x_4) = (3713, 103, 6, 2)$*, and four residues have binary representation as* $x_1 = 3713 = (111010000001)_2$*,* $x_2 = 103 = (1100111)_2$*,* $x_3 = 6 = (110)_2$*, and* $x_4 = 2 = (10)_2$*. We can convert the residue number* (x_1, x_2, x_3, x_4) *into its equivalent weighted number* X*. Firstly, using bit organization of systems (8)-(19), we have*

$$v_1 = (0011111)_2 = 31,\ v_{21} = (0001011)_2 = 11,\ \textit{and}\ v_{22} = (1111011)_2 = 123.$$

$$H = |v_1 + v_{21} + v_{22}|_{2^7-1} = |31 + 11 + 123|_{2^7-1} = |165|_{127} = 38.$$

$$v_3 = (001)_2 = 1,\ v_4 = (100)_2 = 4,\ \textit{and}\ K = |v_3 + v_4|_{2^3-1} = |1 + 4|_{2^3-1} = 5.$$

$$v_5 = (000110)_2 = 6,\ v_6 = (101101)_2 = 45,\ v_{71} = (0000101)_2 = 5,$$

$$v_{72} = (111110)_2 = 62,\ v_{81} = (0011001)_2 = 25,\ v_{82} = (111111)_2 = 63,$$

$$\textit{and}\ T = |v_5 + v_6 + v_{71} + v_{72} + v_{81} + v_{82}|_{2^6-1} = 17.$$

Next, the above values are substituted for system (20), so

$$S = H + 2^{n+1}T - T = 38 + 128 \times 17 - 17 = 2197.$$

Then, X *can be evaluated as*

$$X = x_1 + 2^{2n}S = 3713 + 4096 \times 2197 = 9002625.$$

To verify the result, we have

$$x_1 = |9002625|_{4096} = 3713,\ x_2 = |9002625|_{127} = 103,$$

$$x_3 = |9002625|_9 = 6,\ x_4 = |9002625|_7 = 2.$$

The weighted number X *based on the 4-moduli set has expression as* $(3713, 103, 6, 2)$.

3. **Hardware Architecture Design.** Figure 1 shows the new architecture design for four-moduli set $\{2^{2n}, 2^{n+1}-1, 2^{n/2}+1, 2^{n/2}-1\}$ with resides $\{x_1, x_2, x_3, x_4\}$ input. In the first stage, the function block of bit organizer 1 reorders input to obtain v_1, v_{21}, v_{22}, v_3 and v_4. This function is required $(1.5n-2)$ NOT gates. Next, CSA 1 (carry save adder), CPA 1 (carry propagation adder), and CPA 2 are used to perform H and K. From hardware cost evaluation, three adders take $2.5n$ FA's and 2 HA's. In the second stage, four intermediate values of H, x_1, x_3, and K are reordered by the block of bit organizer 2 to produce v_5 to v_{82}. It uses $(3n+1)$ NOT gates for data inverse. Then, CSA 2 to 5 and CPA 3 are used for calculating T. They need to spend $(3.5n+2)$ FA's and $(1.5n-2)$ HA's totally. In the third stage, bit organizer 3 reorders T and H to express $H + 2^{n+1}T$ and $\overline{T}$. This function is needed n NOT gates. Subsequently, we use CPA 4 to sum two values of $H + 2^{n+1}T$ and $\overline{T}$ to obtain the output S. It takes n FA's and $(n+1)$ HA's. Next, according the critical path, the total delay of the proposed architecture is $(1+2n+2+1+1+1+2n+2n+1)t_{\mathrm{FA}} = (6n+7)t_{\mathrm{FA}}$ where t_{FA} represents the operation delay of 1-bit full adder.

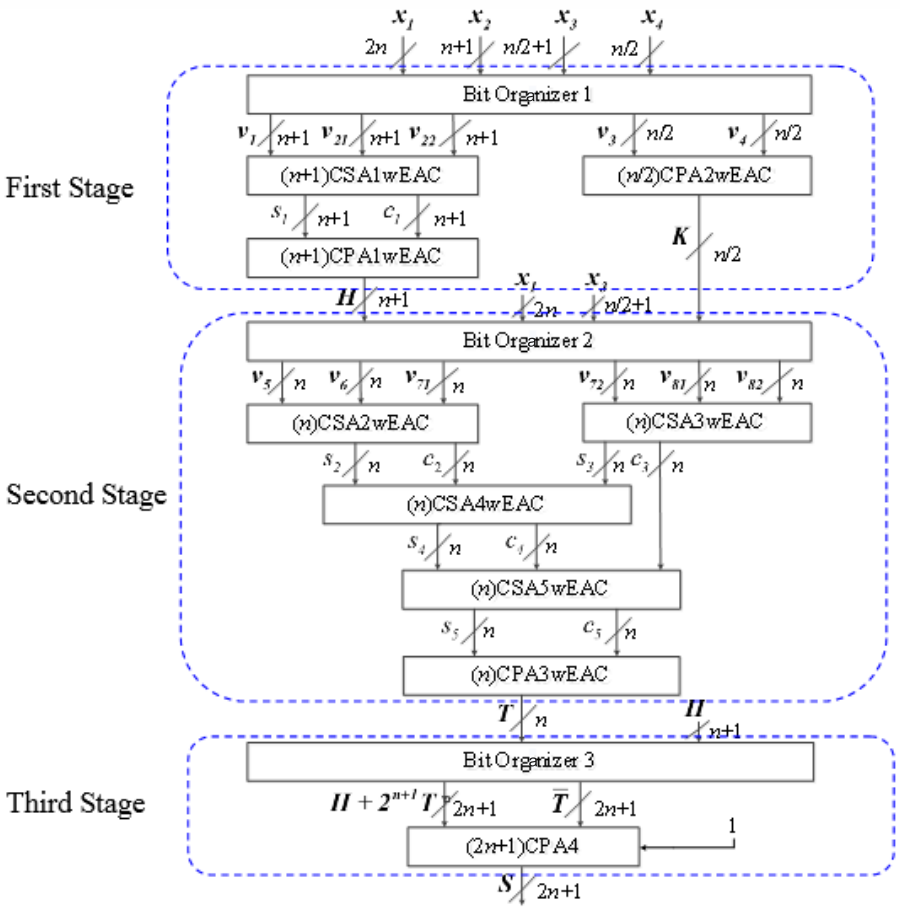

FIGURE 1. The new converter for 4-moduli set $\{2^{2n}, 2^{n+1}-1, 2^{n/2}+1, 2^{n/2}-1\}$

4. **Performance Evaluation.** Table 1 lists converter performance comparison between our design and other works. It shows the proposed design has the shortest critical delay. Our design and [10] can achieve the lowest hardware costs. Based on TSMC 90 nm CMOS technology, the proposed converter and the other converters [10,11] are synthesized by using Synopsys and Cadence tools. Table 2 lists the results of layout areas, powers, and

TABLE 1. Converter performance comparison

Converter	Hardware cost	Critical delay
[1]	$(7n+2)A_{\mathrm{FA}} + (4n+2)A_{\mathrm{HA}} + (6.5n+2)A_{\mathrm{INV}}$	$(8n+7)t_{\mathrm{FA}} + 3t_{\mathrm{NOT}}$
[11] C-I	$(13n+2)A_{\mathrm{FA}} + (6n+1)A_{\mathrm{HA}} + (8n)A_{\mathrm{NOT}}$	$(8n+1)t_{\mathrm{FA}} + 6t_{\mathrm{NOT}}$
[11] C-II	$(16n+1)A_{\mathrm{FA}} + (7n+1)A_{\mathrm{HA}} + (11n)A_{\mathrm{NOT}}$	$(8n+2)t_{\mathrm{FA}} + 6t_{\mathrm{NOT}}$
[12]	$(2n^2+11n+3)\,A_{\mathrm{FA}} + (7n-1)A_{\mathrm{HA}}$	$(11.5n + 2\log_2 2n + 2.5)\,t_{\mathrm{FA}}$
Proposed	$(7n+2)A_{\mathrm{FA}} + (3.5n+1)A_{\mathrm{HA}} + (5.5n-1)A_{\mathrm{INV}}$	$(6n+7)t_{\mathrm{FA}} + 3t_{\mathrm{NOT}}$

TABLE 2. Each converter layout and comparison result

	DR = 32 bits			DR = 64 bits			DR = 96 bits		
Converter	Area (μm^2)	Power (mW)	Delay (nS)	Area (μm^2)	Power (mW)	Delay (nS)	Area (μm^2)	Power (mW)	Delay (nS)
[1]	4912.3	5.7	13.3	6998	8.8	23.8	10916.1	13.3	33.2
[11] C-I	6112.6	7.5	11.6	8474.2	10.8	19.2	12989.3	16.9	28.3
[11] C-II	5610.2	7.2	10.3	9516.4	12.3	17.8	13809.2	17.2	23.4
Proposed	4807.9	5.6	9.4	6474.5	8.5	17.1	9993.4	12.8	21.6

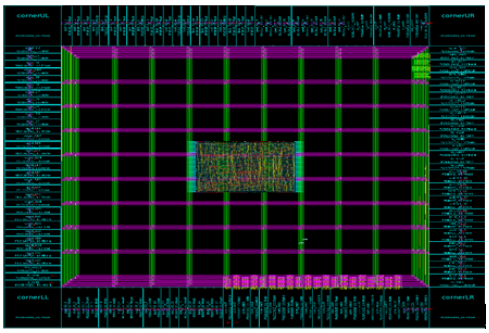

FIGURE 2. VLSI implementation for the proposed 64-bit reverse converter

delays from 32- to 96-bit width. The proposed converter and [10] can achieve the smallest area and power, but our design can further improve 18.6% on average saving in critical delay. Under the same DR, our converter has over 32.1% and 21.8% saving in ADP, and AD, respectively. In Figure 2, the chip layout implementation of the 64-bit ($n = 16$) reverse converter core area is $239.2 \times 239.7\mu\text{m}^2$. When considering the parasitic effects of wire loading and I/O pad, the working frequency is about 51.5MHz, and the power consumption is measured at 2.7mW when using a 0.9V voltage supply.

5. **Conclusions.** This paper presents a novel reverse conversion algorithm for 4-moduli set $\{2^{2n}, 2^{n+1}-1, 2^{n/2}+1, 2^{n/2}-1\}$ that has $4n$-bit DR. This new method can derive an efficient reverse converter with shorter bit-width CPA which can promote operation speed dramatically. From experimental results, the proposed converter has better performance with respect to APD, AD measurements when compared with the latest 4-moduli sets. Finally, a 64-bit reverse converter is implemented on VLSI which can be profitable for constructing a RNS-based DSP system. Because reverse converter has the longest operation time in the whole system, promotion of its conversion time is necessary to be researched in the future.

REFERENCES

[1] D. Zivaljevic, N. Stamenkovic and V. Stojanovic, Digital filter implementation based on the RNS with diminished-1 encoded channel, *IEEE Intl. Conf. on Telecommunications and Signal Processing*, pp.662-666, 2012.

[2] F. T. Thian and C. H. Chang, A new algorithm for single residue digit error correction in redundant residue number system, *IEEE Int. Symp. Circuits and Systems*, pp.1748-1751, 2014.

[3] G. Perin, L. Mbert, L. Torres and P. Maurine, Electromagnetic analysis on RSA algorithm based on RNS, *IEEE Euromicro Conf. Digital System Design*, pp.345-352, 2013.

[4] L. Sousa and P. Martins, Efficient sign identification engines for integers represented in RNS extended 3-moduli set $\{2^n-1, 2^{n+k}, 2^n+1\}$, *IET Electronics Letters*, vol.50, pp.1138-1139, 2014.

[5] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Design*, 2nd Edition, Oxford University Press, New York, 2010.

[6] Y. Wang, X. Song, M. Aboulhamid and H. Shen, Adder based residue to binary numbers converters for $(2^n-1, 2^n, 2^n+1)$, *IEEE Trans. Signal Processing*, vol.50, no.7, pp.1772-1779, 2002.

[7] A. S. Molahosseini, M. K. Rafsanjani, S. H. Ghafouri and M. Hashemipour, An improved RNS reverse converter for the $\{2^{2n+1}-1, 2^n, 2^n-1\}$ moduli set, *Proc. of IEEE Int. Symp. on ISCAS*, pp.2103-2106, 2010.

[8] M. Esmaeildoust, K. Navi and M. R. Taheri, A new five-moduli set for efficient hardware implementation of the reverse converter, *IEICE Electron. Exp. Letters*, vol.6, no.14, pp.1006-1012, 2009.

[9] M. Esmaeildoust, K. Navi and M. R. Taheri, High speed reverse converter for new five-moduli set $\{2^n, 2^{2n+1}-1, 2^{n/2}-1, 2^{n/2}+1, 2^n+1\}$, *IEICE Electron. Exp. Letters*, vol.7, no.3, pp.118-125, 2010.

[10] S. Rizazi, S. Hassanpour and M. Hosseinzadeh, An efficient architecture of residue to binary converter for new four-moduli set, *Int. J. Comp. Tech. Appl.*, vol.2, no.4, pp.709-715, 2011.

[11] L. Sousa and S. Antao, MRC-based RNS reverse converter for the four-moduli sets $\{2^n+1, 2^n-1, 2^n, 2^{2n+1}-1\}$ and $\{2^n+1, 2^n-1, 2^{2n}, 2^{2n+1}-1\}$, *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol.59, no.4, pp.244-248, 2012.

[12] L. Sousa, S. Antao and R. Chaves, On the design of RNS reverse converters for the four-moduli set $\{2^n+1, 2^n-1, 2^n, 2^{n+1}+1\}$, *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol.21, no.10, pp.1945-1949, 2013.