Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 8, No. 4, November 2017), Pages: 87-94
www.jacr.iausari.ac.ir

Efficient Reverse Converter for Three Moduli Set $\{2^n-1, 2^{n+1}-1, 2^n\}$ in Multi-Part RNS

Shiva Taghipour Eivazi✉
Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran
taghipour@iaut.ac.ir

Received: 2016/10/17; Accepted: 2017/01/07

Abstract

Residue Number System is a numerical system which arithmetic operations are performed parallelly. One of the main factors that affect the system's performance is the complexity of reverse converter. It should be noted that the complexity of this part should not affect the earned speed of parallelly performed arithmetic unit. Therefore in this paper a high speed converter for moduli set $\{2^n-1, 2^{n+1}-1, 2^n\}$ *is proposed which is based on Two-Part RNS and Chinese Reminder Theorem. Using this method has increased the speed of reverse converter. To have an accurate comparison both unit gate model and synthesized silicon tools are used and their parameters are compared in terms of delay and area. Converters are implemented in hardware description language and correctness for various n values is verified by simulation and execution on Cadence. As the results show, the proposed circuit has lower delay by around 21% in comparison to previous presented converter.*

Keywords: *Chinese Remainder Theorem (CRT), Computer Arithmetic, Parallel Processing, Residue Number System (RNS), R/B Converter, VLSI Architectures*

1. Introduction

Residue number system (RNS) is a numerical system which is able to perform the arithmetic operations parallelly on remainders of the selected moduli set. Because the residues are smaller than the original numbers and due to the carry free nature of RNS, arithmetic operations are performed fast [1], [2]; therefore this system is suitable for applications that perform arithmetic operations such as addition, subtraction and multiplication widely on a special boundary such as Digital Signal Processing (DSP) [3], image processing [4], [5] and digital filters [6], [7].

To use RNS, numbers should be converted into the residue representation following the arithmetic operations. Then the resulting numbers are converted to the desired system.

By the above mentioned description, the converter`s operations are the most overloaded computation of RNS that will lead to increase of time complexity. Fortunately forward converter could be implemented simply by using modular adders [8]. But backward converter is the most complex step which burdens a high time complexity to the system. To decrease the overload of reverse converter using multi-part RNS is suggested in this paper.

2. Multi-Part RNS

Consider number (A) in binary representation, be represented by $\{a_{m-1}, a_{m-2}, ..., a_0\}$.

To convert this number to multi-part RNS, the original number is partitioned into some parts. Each of which requires a corresponding and independent moduli sets. For example the number below is partitioned into two parts.

$$\underbrace{\{a_{m-1}, a_{m-2}, ..., a_n\}}_{part(a)\, m-n bit}\underbrace{\{a_{n-1}, a_{n-2}, ..., a_0\}}_{part(b)\, n bit}$$

The following steps are performed for each part:

The forward converter is implemented parallelly.

The arithmetic operations are performed on each part's moduli set. Note that the output carry of each part should be carried out to the next.

The backward converter of each part is defined independently.

Due to the second part, it is sufficient to implement a two-part RNS, leading limited carry propagation among separated parts, as it is mentioned in [9].

Consider the moduli set $\{2^n-1, 2^{n+1}-1, 2^n\}$ [10]. The dynamic range of this moduli set in usual RNS is calculated as follows:

$$M = \prod_{i=1}^{3} m_i = (2^n-1)\times(2^{n+1}-1)\times 2^n = 2^{3n+1} - 2^{2n+1} - 2^{2n} + 2^n \quad (1)$$

To implement the two-part RNS, it is adequate to keep the n lowest value bits in one part via moduli $\{2^n\}$ and convert the other 2n+1 most value bits to RNS via moduli set $\{2^n-1, 2^{n+1}-1\}$ for the next part. Let $0 \le X < M$ be with binary representation. By applying two-part RNS for moduli set $\{2^n-1, 2^{n+1}-1, 2^n\}$ we have:

$$X = \left| \underbrace{x_{3(n)}x_{3(n-1)}...x_{n+1}x_n}_{\substack{part(a)\\ \{2^n-1, 2^{2n+1}-1\}}} \underbrace{x_{n-1}x_{n-2}...x_1x_0}_{\substack{part(b)\\ \{2^n\}}} \right|_2$$

Then the numbers of modulis are reduced to two in part(a), which leads less complexity of converters [8].

Example 1:

Consider weighted number 620 that is $(1001101100)_2$ in binary form and given moduli set is $\{2^n-1, 2^{n+1}-1, 2^n\}$ where n=3, therefore modulis are {7, 15} for part(a) and {8} for part(b). 620 is represented in two-part RNS as $\left| \underbrace{1001101}_{\substack{part(a)=77\\ module\ set\ \{7,15\}}} \underbrace{100}_{\substack{part(b)=4\\ module\ set\ \{8\}}} \right|_2 = \{0_{\bmod 7}, 2_{\bmod 15}\} \mid \{4_{\bmod 8}\}$.

Moreover, multi part system increases the dynamic range which can be calculated as follows:

$$M(a) = (2^n-1)\times(2^{n+1}-1) = 2^{2n+1} - 2^n + 1 = \underbrace{11...1}_{n-1}0\underbrace{10...0}_{n-1}1 \quad (2)$$

As mentioned before, this number creates the high order 2n+1 bit. The lowest value n bits comes from the second part. Therefore final dynamic range is calculated as follows:

$$M_{total} = \underbrace{(2^{2n+1} - 2^n + 1)}_{part(a)} \times 2^n + \underbrace{2^n}_{part(b)} \quad (3)$$

Thus it increases dynamic range by 2^n.

3. Suggested Reverse Converter

For two-part RNS, two backward converters are required to form original number as follow:

A backward converter for moduli set $\{2^n - 1, 2^{n+1} - 1\}$ which forms the most 2n+1 bits value.

A backward converter for the moduli set $\{2^n\}$ to form the n low bits.

3.1 Reverse converter for part (a) with moduli set $\{2^n - 1, 2^{n+1} - 1\}$ *:*

The moduli in first part reduce to $\{2^n - 1, 2^{n+1} - 1\}$ where $< x_1, x_2 >$ stand the reminder of each moduli respectively.

Both Chinese remainder theorem (CRT) [8] and mixed-radix conversion (MRC) [11] are algorithms that are used to implement a converter. In this paper CRT is used as (4).

$$X = \left\langle \sum_{i=1}^{n} (X_i \times N_i)_{m_i} \times M_i \right\rangle_M \quad (4)$$

Where:

$$M = \frac{\prod_{i=1}^{n} m_i}{GCD(m_i, m_j), i \neq j} \quad (5)$$

$$M_i = \frac{M}{m_i} \quad (6)$$

$$N_i = < M_i^{-1} >_{m_i} \quad (7)$$

Due to the number of moduli in first part, a two channel CRT should be used [12].M is equal to dynamic range of part(a) that is calculated in (2). As (6), M_i for each modulus is equal to:

$$M_1 = \frac{(2^n - 1) \times (2^{n+1} - 1)}{2^n - 1} = 2^{n+1} - 1 \quad (8)$$

$$M_2 = \frac{(2^n - 1) \times (2^{n+1} - 1)}{2^{n+1} - 1} = 2^n - 1 \quad (9)$$

The multiplicative inverse for each moduli is as (10) and (11).

$$N_1 =< M_1^{-1} >_{2^{n+1}-1} = 1 \tag{10}$$

$$N_2 =< M_2^{-1} >_{2^n-1} = 2^{n+1} - 3 \tag{11}$$

Therefore X for moduli set $\{2^n - 1, 2^{n+1} - 1\}$ is calculated as follow:

$$X(a) = \left\langle ((x_1 \times N_1)_{2^n-1} \times (2^{n+1} - 1)) + ((x_2 \times N_2)_{2^{n+1}-1} \times (2^n - 1)) \right\rangle_{M(a)} \tag{12}$$

$$X(a) = \left\langle (x_1 \times (2^{n+1} - 1)) + ((x_2 \times 2^{n+1} - 3)_{2^{n+1}-1} \times (2^n - 1)) \right\rangle_{M(a)} \tag{13}$$

Where $\left\langle (x_2 \times 2^{n+1} - 3)_{2^{n+1}-1} \right\rangle_{M(a)}$ is equal to $|(\overline{x_{2(n-1)} x_{2(n-2)}} \ldots \overline{x_{2(1)} x_{2(n)}})|_{M(a)}$. By replacing (13) we have:

$$X(a) = \left\langle (x_1 \times (2^{n+1} - 1)) + (\overline{x_{2(n-1)} x_{2(n-2)}} \ldots \overline{x_{2(1)} x_{2(n)}}) \times (2^n - 1)) \right\rangle_{M(a)} \tag{14}$$

$$X(a) = \left\langle (x_1 \times 2^{n+1}) - x_1 + (\overline{x_{2(n-1)} x_{2(n-2)}} \ldots \overline{x_{2(1)} x_{2(n)}} \times 2^n) - \overline{x_{2(n-1)} x_{2(n-2)}} \ldots \overline{x_{2(1)} x_{2(n)}} \right\rangle_{M(a)} \tag{15}$$

By using (15) implementation of reverse converter is performed as follow. Note that two's complement is applied to perform the subtraction.

$$A = \underbrace{x_{1(n-1)} x_{1(n-2)} \ldots x_{1(0)}}_{n} \underbrace{x_{2(n-1)} x_{2(n-2)} \ldots x_{2(0)} x_{2(n)}}_{n+1} \tag{16}$$

$$B = \underbrace{\overline{x_{2(n-1)} x_{2(n-2)}} \ldots \overline{x_{2(0)} x_{2(n)}}}_{n+1} \underbrace{\overline{x_{1(n-1)} x_{1(n-2)}} \ldots \overline{x_{1(0)}}}_{n} \tag{17}$$

$$C = \underbrace{11\ldots1}_{n-1} 10 1 \underbrace{00\ldots0}_{n-2} 10 \tag{18}$$

As seen A, B, C are with 2n+1 bits. The calculations are as following:

$$X(a) = |A + B + C|_{M(a)} \tag{19}$$

The below steps demonstrate the new residue to binary converter algorithms.

Calculate $X(a) = |A + B + C|$

If $(X(a) < M_a)$ stop. (The result obtained)

Otherwise $(X(a) - M_a)$ is as output.

3.2 Reverse converter for part(b) with modulus set $\{2^n\}$ *:*

x_3 is the reminder of moduli $\{2^n\}$ in part(b). As known, to implement the reverse converter for moduli $\{2^n\}$ no hardware implementation is required and choosing the n lowest order bits of x_3 is sufficient [12].

Finally, the resulted X for two-part RNS via moduli set $\{2^n - 1, 2^{n+1} - 1, 2^n\}$ can be obtained from (20).

$$X = \underbrace{X(a)}_{\substack{part(a) \\ (2n+1)bit}} \times 2^n + \underbrace{X(b)}_{\substack{part(b) \\ (n)bit}} \tag{20}$$

4. Hardware Implementation

Fig1 illustrates the hardware implementation of the proposed residue to binary converter in two-part RNS. Due to (16-18), A, B and C are generated. As the first part of above algorithm (A+B+C) should be calculated which is first done by the 2n+1 bit Carry Save Adder (1) then the resulted 2n+1 bit carry and sum, should be added by carry propagate adder (1).

To implement the modular adder in part(a) as third step of algorithm, the subtraction is implemented by 2's complement adder, therefore 2n+1 bit carry save adder (CSA 2) is applied to add the resulted carry1 and sum1 with two's complement of M(a) which is named as D in Fig1 and it is equal to $\underbrace{00...0}_{n-1}1\underbrace{011...1}_{n}$.

Then the resulted 2n+1 bit carry2 and sum2 should be added that is done by carry propagate adder (2).The output carry of CPA (2) determines which result is valid for the part (a).

Finally the n low bits from part (b), form the final result's n lowest bits is generated.

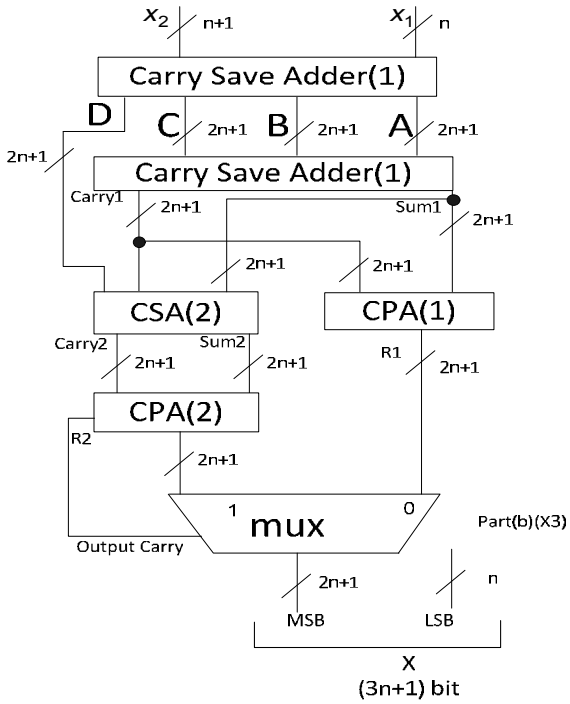

Figure 1. Proposed reverse converter for moduli set $\{2^n-1, 2^{n+1}-1, 2^n\}$

Example 2:

Consider n=3, given residues are $< x_1, x_2 >| x_3 =< 3_{\mathrm{mod}\,7}, 4_{\mathrm{mod}\,15} >| 5_{\mathrm{mod}\,8}$ respectively. As (16-18) we have:

$A = 0111000, B = 0111100, C = 1101010$

Due to Fig1 $carry1 = 1110000$ and $sum1 = 1101110$. Therefore $R1 = 1011110$.By adding two's complement of $M(a)$ R2 is generated that is $R2 = \underbrace{0}_{outputCarry}\ 1110110$. Due to the output carry, R1 is chosen in this example. Therefore the final X is formed as $X = (1110110101)_2$. Where $(101)_2$ is x_3.

Example 3:

Again consider $n=3$. The given residues are $<x_1,x_2>|x_3=<1_{\mathrm{mod}7},0_{\mathrm{mod}15}>|2_{\mathrm{mod}8}$. Therefore:

$A=0010000, B=1111110, C=1101010$

Due to Fig1 $carry1=1110100$ and $sum1=0000100$. We have $R1=1111000$.

By adding two's complement of $M(a)$ R2 is generated that is $R2=\underset{outputCarry}{\underbrace{1}}\ 0001111$. Due to the output carry value, R2 is chosen as output. Finally X is formed as $X=(0001111010)_2$. Where $(010)_2$ is x_3.

5. Comparison

Recently presented reverse converters for moduli set $\{2^n-1,2^{n+1}-1,2^n\}$ have been introduced in [13-15]. Note that in this paper a FA with constant input ("0" or "1") is considered as HA. Due to table 1, in comparison to the listed reverse converters, the proposed circuit in this paper has lower delay.

Table 1. Performance Comparison

Converter	INV	MUX	ROM	HA	FA	Delay
Proposed	2n+1	2n+1	-	4n+4	4n	$t_{not}+t_{mux}+3_{tHA}+(2n)t_{FA}$
[15]	4n+1	-	-	2n+1	4n+1	$(6n+5)t_{FA}$
[14]	3n+2	n+1	-	2n+1	5n+3	$2t_{not}+t_{mux}+(n+1)t_{HA}+(3n+4)t_{FA}$
[13]C-I	5n+4	-	-	n	4n+3	$4t_{not}+nt_{HA}+(5n+5)t_{FA}$
[13]C-II	5n+2	2n+1	-	2n+3	14n+21	$t_{not}+t_{mux}+2t_{HA}+(2n+4)t_{FA}$
[13]C-III	5n+2	2n+1	10×(2n+1)	2n+2	12n+19	$t_{not}+t_{mux}+2t_{HA}+(2n+4)t_{FA}$

To provide a fair comparison the proposed circuit is compared with the previous implementations by considering unit gate model which is used in [16, 17]. The results are reported in Table 2.

Table 2. Total Unit Gate Area and Delay of the Reverse Converters based on unit gate model

Converters	Unit Gate Area	Unit Gate Delay	Time Complexity
Proposed	52n+20	8n+9	416 n^2+628 n+ 180
[15]	40n+12	24n+10	960 n^2+1488 n+ 360
[14]	49n+30	14n+22	686 n^2+1498 n+ 660
[13]C-I	37n+25	22n+24	814 n^2+1438 n+ 600
[13]C-II	123n+167	8n+25	984 n^2+4411 n+ 4175
[13]C-III	103n+146	8n+23	824 n^2+3204 n+ 3537

As it is shown in table2, the proposed converter in [14] is the best design among previous methods and the suggested two-part method in this paper is better than [14] by 1.64 times. Therefore the proposed design and the converter in [14] are implemented and the correctness of two converters is checked by running on the corresponding VHDL codes. Furthermore, we have used these codes to synthesize (Design vision) and then Cadence (SOC encounter) using a target library based on TSMC 0.18 μm. The working frequency is 155 MHZ where voltage supply is considered 1.8V.

Table 3. Synthesizes results

Moduli set	Converter	Area(μm2)	Delay(ns)	Min working period(ns)
{3,7,4}	Proposed	580*580	1.16	2
	[14]	550*520	1.32	5
{15,31,16}	Proposed	700*700	1.23	2.5
	[14]	640*610	1.40	6
{31,63,32}	Proposed	760*760	1.25	2.5
	[14]	670*670	1.41	6
{255,511,256}	Proposed	940*940	1.51	2.5
	[14]	820*790	1.90	6.5

The performance evaluation has been done in terms of both area and delay (with Minimum working period). Table 3 has presented the earned results. As shown in the table the proposed design is the fastest design for any moduli set

The reason of this improvement is related to using multi-part method that facilitates the implementation of the proposed converter and also the simple architecture of proposed converter that is mainly consists of adders which offers lower Area and Delay metric in comparison previous reverse converters.

The final layout of proposed converter for n=8 is shown in Figure 2.

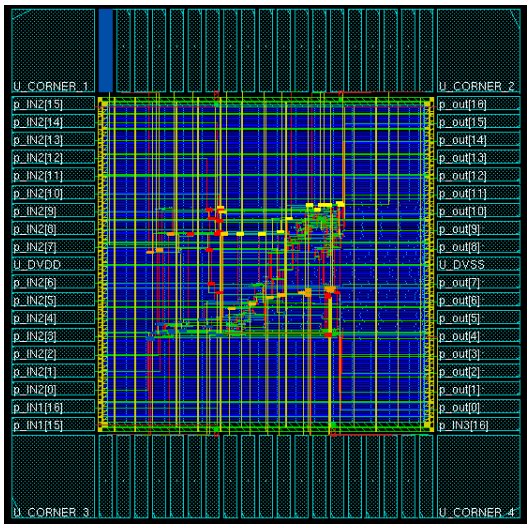

Figure 2. Chip layout of proposed backward converter for n=8.

6. Conclusion

In this paper a high speed low complexity reverse converter have been proposed for moduli set $\{2^n-1, 2^{n+1}-1, 2^n\}$. The novel converter has been designed in two-part structure and in parallel which results in increased conversion speed. To have an accurate comparison both unit gate model and simulation are used. As the results of unit gate illustrates the proposed method indicates an improvement of about 1.68 time complexity. Also, the novel design is compared accurately with the related state of the art design by running the corresponding VHDL codes with cadences. As the results indicate, for any n, the proposed converter is improved in delay point because of using multi-part approach.

References

[1] A. Hiasat, "A Reverse Converter and sign detectors for an Extended RNS Five-Moduli set," IEEE Transactions on Circuits and Systems I, Regular Papers, 2017.

[2] Sh. Taghipour, M. Hosseinzadeh, and O. Mirmotahhari, "Fully Parallel Comparator for the moduli set $\{2^n, 2^n-1, 2^n+1\}$," IEICE Electronic Express, 2011.

[3] G. Cardarilli, A. Nannarelli, and M. Re, "Residue number system for low-power DSP applications," in Proc. 41st IEEE Asilomar Conference Signals, System, Computer, 2007.

[4] W. Wei, M.N.s. Swamy, and M. O. Ahamd, "RNS application for digital image processing," in Proc. 4th IEEE International Workshop System on Chip Real Time Applications, 2004.

[5] A. Ammar, A. AlKabbany, M. Youssef, and A. Emam, "A secure image coding using residue number systems," In Proccessing, 18th national Radio Science Conference, 2001.

[6] R. Conway, and J. Nelson, "Improved RNS FIR filter architectures," IEEE Transactions on Circuits Systems II, Express Briefs, 2004.

[7] V. Singh, "Improved state-space criterion for global asymptotic stability of fixed-point state-space digital filters with saturation arithmetic," The Arabian Journal for Science and Engineering, 2007.

[8] K. Navi, A. Mollahosseini, and M. Esmaeildoust, "How to Teach Residue Number System to Computer Scientists and Engineers," IEEE Transaction on Education, 2011.

[9] B. Kazemzadeh, and Sh. TaghipourEivazi, "High Speed Comparator by using two-Part RNS," International Journal of Computer Science and Information Security, 2016.

[10] Sh. TaghipourEivazi, M. Hosseinzadeh, and A. HabibiZadnavin, "Efficient RNS Converter via Two-Part RNS," Journal of Circuits, Systems, and Computers, 2015.

[11] B. Cao, C. Chang, and T. Srikanthan, "Adder based residue to binary converters for a new balanced 4-moduli set," Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis, 2003.

[12] M. Jameii, Sh. Taghipoureivazi, and M. azad, "Using both Binary and Residue Representations for Achieving Fast Converters in RNS," Journal of Advances in Computer Research, 2011.

[13] P.V.A Mohan, "RNS-to-binary converter for a new three-moduli set $(2^{n+1}-1, 2^n, 2^n-1)$," IEEE Transactions on Circuits and Systems II, 2007.

[14] S. LIN, M. SHEU, and C. Hsiang, "Efficient VLSI Design of Residue-to-Binary Converter for the Moduli set $(2^n, 2^{n+1}-1, 2^n-1)$," IEICE Transactions on Information System, 2008.

[15] Y. Chingkuo, M. Siao, Ch. Huang, and T. Chen, "New Reverse Converter Design of Moduli Set $\{2^n, 2^{n+1}-1, 2^n-1\}$," Second International Conference on Innovations in Bio-inspired Computing and Applications, 2011.

[16] M. R. Noorimehr, M. Hosseinzadeh, and R. Farshidi, "High Speed Residue to Binary Converter for the New Four-Moduli Set $\{2^{2n}, 2^n+1, 2^{n/2}+1, 2^{n/2}-1\}$," Arabian Journal of Science and Engineering, 2014.

[17] M. R. Noorimehr, M. Hosseinzadeh, and K. Navi, "Efficient reverse converters for 4-moduli sets $\{2^{2n-1}-1, 2^n, 2^n+1, 2^n-1\}$ and $\{2^{2n-1}-1, 2^{2n-1}, 2^n+1, 2^n-1\}$ based on CRTs algorithm," Circuits Systems and Signal Processing, 2014.