IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 2, Issue 6 (Jul. – Aug. 2013), PP 14-19
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Design of RNS Converters for moduli sets with Dynamic Ranges up to 6*n*-bit

Shubham Kaushik, Ashish Srivastava
M.Tech- VLSI, VIT University, Vellore, Tamil Nadu, India

Abstract: *The RNS has been considered as an interesting area for researchers in recent year. This paper presents memoryless and area efficient RNS converters for moduli set with dynamic ranges up to 6n-bit. Residue number system (RNS) has mainly targeted parallelism and larger dynamic ranges. In this paper, we start from the moduli sets* $\{2^n, 2^n - 1, 2^n + 1, 2^n - 2^{(n+1)/2} + 1, 2^n + 2^{(n+1)/2} + 1\}$ *with dynamic range of 5n-bit and propose vertical extension in order to improve the parallelism and increase the dynamic range. The vertical extension increase the value of the power of 2 modulus in the five-moduli set. The Chinese remainder theorem is applied in this paper to derive an efficient reverse converter. This paper also proposed a conventional binary to RNS representation called RNS Forward conversion. The RNS forward converter is more efficient in terms of area, delay and power. Synthesis results suggest that the proposed vertical extension in RNS reverse converter allow reducing the area-delay product in comparison with the related state-of-the-art.*

Keywords- *Chinese Remainder theorem, Forward converter, Residue Number system, Reverse converter*

I. Introduction

The conventional 2's complement number system imposes a fundamental limitation on power-performance efficiency as a result of sequential carry propagation. The residue number system (RNS) breaks this limitation by portioning operation into parallel independent components resulting in fast and power efficient hardware. The arithmetic operations such as addition, subtraction, and multiplication, can be carried out independently and concurrently in several residue channels more efficiently than in the conventional 2's complement binary system. The adoption of the RNS has provided significant efficiency for different types of Digital signal processing (DSP) applications such as filtering, computation of the discrete cosine transform and cryptography.

The general structure of RNS based processor is shown in Fig 1. It includes both forward as well as reverse converter.

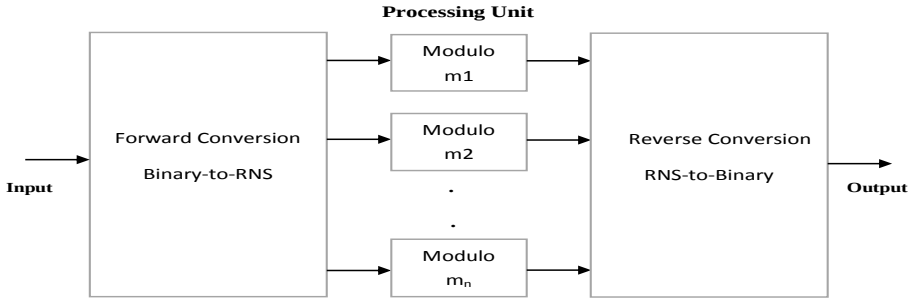


Fig. 1 General Architecture of RNS-based processor

The process of converting the data from binary to residue number system representation is called forward conversion. The forward converter is an efficient converter in terms of area, delay and power. On the other hand, the reverse conversion involves the conversion of the data from RNS representation to conventional binary representation. The design of RNS reverse converter is the most important step for any successful RNS. In this paper, we focus on a vertically extended area efficient RNS converter and a conventional forward converter and their comparison on the basic of area, delay and power.

The algorithm for reverse conversion are mainly based on the Chinese Remainder Theorem (CRT), on the mixed-radix conversion (MRC), and on what has more recently been called the New Chinese Remainder Theorems (New CRTs) [1].

The rest of the paper is organized as follows. The RNS Reverse converter is proposed in first section. The proposed vertical extension to the original moduli set is presented in second section. A conventional forward

converter for binary to RNS representation in third section. The proposed reverse converters compared with the related state-of –the-art in the fourth section. The physical design using SOC Encounter in fifth section. Finally, the conclusions are presented in sixth section

II. RNS Reverse Converter

Reverse conversion from RNS to binary representation is based on either Chinese Remainder theorem (CRT) or Mixed Radix Conversion method. The MRC is a sequential approach while CRT can be implemented in parallel. In this paper, Chinese remainder theorem is proposed to represent RNS to binary conversion. The mathematical representation of the CRT is given in the section 3.

1.2 Selection of moduli set

The choice of the moduli set is of key importance in order to obtain balanced moduli sets that exploit parallelism for the Dynamic Ranges (DR) required by the application [1]. The design of reverse converters for these moduli sets is a fundamental issue, because it is a complex and slow operation that has to combine the values of all the residues in order to achieve the equivalent binary representation of the number.

Moduli sets that have been proposed to setup RNS can be classified according to the number of residues and their Dynamic Ranges (DR). Most of the converters for DR around 3n-bit employ moduli of the forms 2^n, $2^n + 1$ and $2^n - 1$. There are some applications such as cryptography, for which the level of parallelism and the DR provided by the three-moduli set are not enough. For those cases, sets with one additional modulus and larger DR have been proposed, as for example the four-moduli sets with a DR of about 4n-bit: **$\{2^n, 2^n - 1, 2^n + 1, 2^{n+1} + 1\}$ [2], $\{2^n, 2^n - 1, 2^n + 1, 2^{n+1} - 1\}$ [2].**

Proposals for DR of 5n-bit can also be found, some of them vertical extensions of the three moduli sets $\{2^n, 2^n + 1, \text{and } 2^n - 1\}$ [2], and four-moduli sets $\{2^n, 2^n - 1, 2^n + 1, 2^{n+1} + 1\}$ [2], $\{2^n, 2^n - 1, 2^n + 1, 2^{n+1} - 1\}$ [1]. In the same direction, of further exploring the vertical extension, the 6n-bit DR moduli sets were recently proposed. In this paper, moduli sets with DR up to 6n-bit is proposed by vertically extending the original moduli set, which is an uncovered task in the related state of the art.

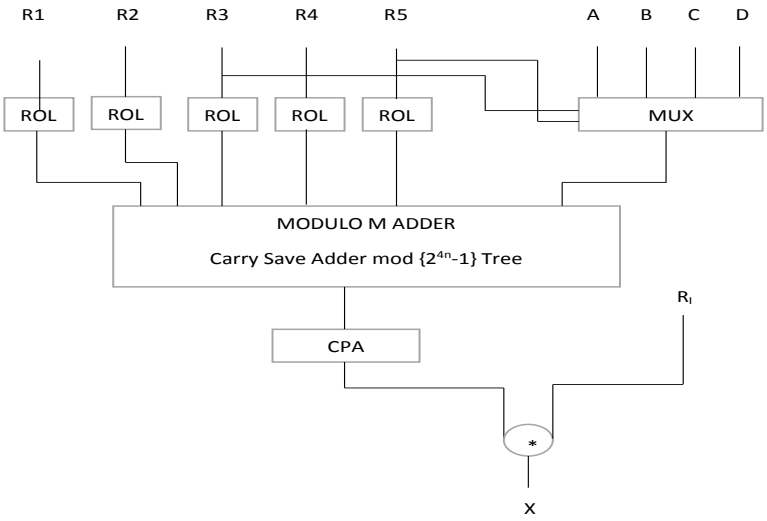


Fig. 2 General architecture of reverse converter

III. Vertical Extension of the Five-Moduli Set

Vertical extension for moduli sets has been proposed to increase the dynamic range of the given moduli sets. The vertical extension increases the value of 2^n modulus by adding some positive integer β in the five moduli set [1]. In order to have a more balanced computation performance and area cost, the value of the exponent of the power of two modulus can be increased by adding an integer positive parameter to it, leading to the five moduli set **$\{2^{n+\beta}, 2^n - 1, 2^n + 1, 2^n - 2^{(n+1)/2} + 1, 2^n + 2^{(n+1)/2} + 1\}$,** with n an odd integer and $n \geq 5$.

To show that this moduli set is composed of co-prime numbers, we only have to show that the modulus $\mathbf{2^{n+\beta}}$ is co-prime with the remaining moduli. This makes the moduli set balanced and leads to simple multiplicative inverses. Naturally, we can apply both the vertical and horizontal extensions, leading to the six-moduli sets. The horizontal extensions of the moduli sets lead to more balanced moduli sets, but the introduction of an additional modulus in the set implies more complex reverse converters. This paper proposed the concept of vertical extension only.

3.1 Proposed Reverse Conversion with Vertical Extension

The extension of the power of two modulus herein proposed adds a parameter β to the exponent, leading to the five moduli set $\{2^{n+\beta}, 2^n - 1, 2^n + 1, 2^n - 2^{(n+1)/2} + 1, 2^n + 2^{(n+1)/2} + 1\}$, with 'n' an odd integer and n≥5. The proposed architecture for the vertically extended five moduli set is shown in Fig 3 [1].

To simplify the presentation of the methods and the description of the architectures herein proposed we adopt the notation followed in [1].

- For an n-bit array of a generic values α_i, bits are referred from the Most Significant Bit (MSB) to the Least Significant Bit (LSB) as $\alpha_{i(n-1)\ldots\ldots}\ \alpha_{i0}$.
- R_i denote the residue for m_i, for which the n-bit array representation is $r_{i(n-1),\ldots,}r_{i0}$.
- The symbol || operates the concatenation of the binary representation of two numbers.

With the dynamic range equal to the product of the moduli for the defined set M, $m_i^{\wedge}$ representing the multiplicative inverse of m_i with respect to the modulus, which are required to decode the RNS representation into binary by using Chinese Remainder theorem (CRT) [1].

The binary representation of the residue numbers is denoted by symbol X, which is mathematically represented as

$$X = \left| \sum m_i^{\wedge} \left| m_i^{\wedge -1} \right| R_i \right|_M$$

It is important to emphasize that the multiplicative inverses consist of a single term for the particular case β = 0, which simplifies the computation of $X/2^n$ and leads to the simple architecture presented in [1]. Each multiplicative inverse contains only one term, hence the architecture will have the same level of complexity as the one presented in [1].

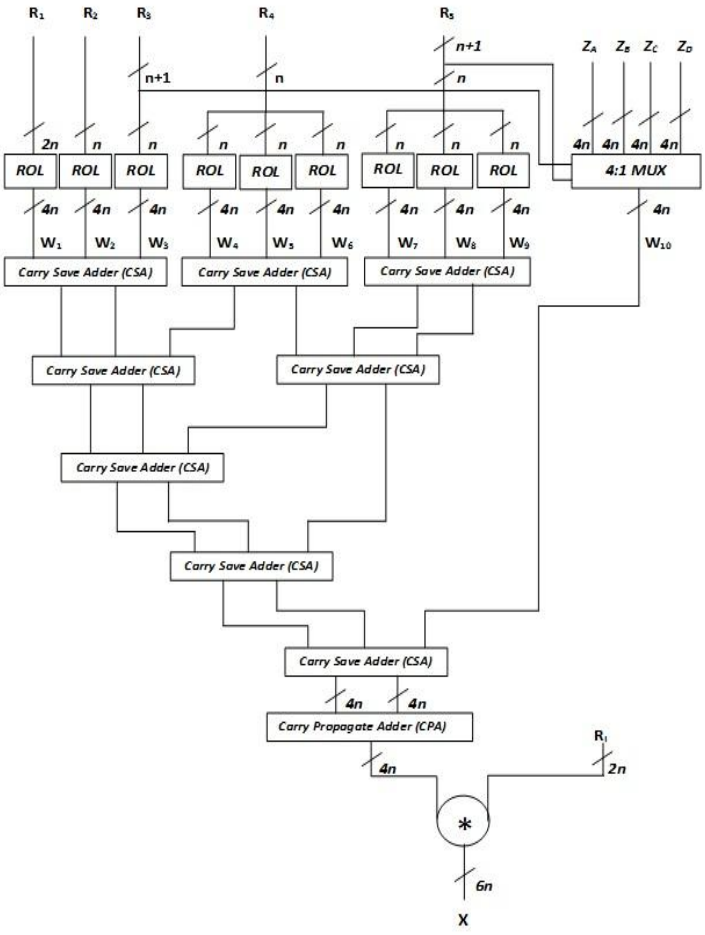


Fig. 4 Proposed RNS reverse converter with vertical extension

IV. RNS Forward Conversion

The forward conversion stage is of paramount importance as it is considered as an overhead in the overall RNS. Choosing the most appropriate scheme depends heavily on the used moduli set. Forward converters are usually classified based on the used moduli into two categories. The first category includes forward converters based on arbitrary moduli-sets. These converters are usually built using look-up tables. The second category includes forward converters based on special moduli-sets. The use of special moduli-sets simplifies the forward conversion algorithms and architectures. The special moduli-set converters are usually realized using pure combinational logic.

We present here some of the available architectures for forward conversion from binary to RNS representation. First, we present forward converters based on arbitrary moduli-sets. Then, we present forward conversion based on the special moduli-set$\{2^n - 1, 2^n, 2^n + 1\}$ [5]. We show how the complexity of the overall design is minimized which reduces the overhead introduced by the forward converter. Finally, we provide an architecture for implementing the modulo addition that are used in the realization of all forward converters.

A typical architecture for the implementation of a forward converter from binary to RNS representation for the special moduli-set $\{2^n - 1, 2^n, 2^n + 1\}$ is shown in Fig. 5 The design of modulo adders is briefly described in the next section. The modulo adder shown in the above Fig.4 is the basic arithmetic unit in RNS operation and converters. Therefore, the performance the modulo adder is very important for the conventional binary to RNS representation.

The modulo adder is one of the basic arithmetic units in RNS operations and converters. The performance of the modulo adder is very critical in the design of forward converters from binary to RNS representation. It is a conventional binary adder that can have different forms such as ripple-carry adder (RCA), carry save adder (CSA). In this paper we used carry save adder to perform modulo operation.

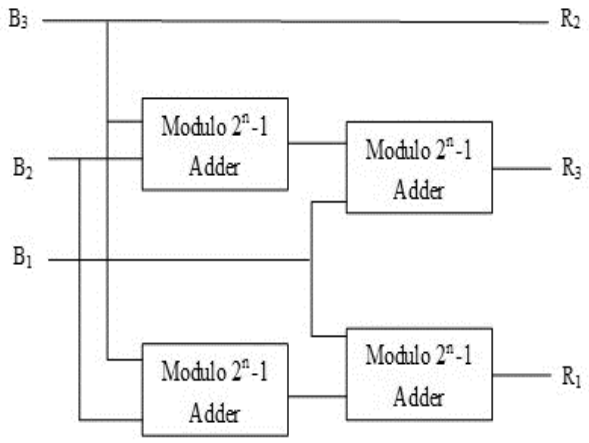


Fig. 5 Proposed RNS forward converter

V. Results and Comparison

The Architectures of the proposed RNS converters are modelled using Verilog HDL and simulation and synthesis is done using the Cadence NCLaunch and RTL compiler tools respectively.

The Synthesis results for the proposed reverse converter are obtained using the 45nm tsmc standard cell library. The obtained results and comparison with the related state-of-the-art are depicted in Table1, showing the obtained area, delay, and power consumption for the achieved dynamic ranges such as 5n-bit and 6n-bit.

The Synthesis results suggest that the proposed vertical extensions allow reducing the area-delay-product up to 4.67 times in comparison with the related state-of-the-art.The Synthesis results for the proposed conventional RNS forward converter for 45nm Tsmc standard cell library are depicted in Table2, showing the obtained area, delay, and power in comparison of RNS reverse converter.

The Synthesis Results suggest that the conventional binary to RNS representation is more area, delay and power efficient than the complex RNS reverse conversion. The proposed Forward converter is allowed to reduce the area-delay product up to 2 times in comparison of proposed reverse converter.

The final chip layout with the specifications of area, density and cells is shown in Fig. 6. The timing reports shown in Table 3 suggest that there is no setup and hold time violations in the design after the detailed placement and routing.

Table1

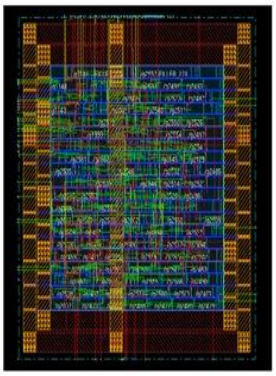
Constraints	Reverse converter [2]	Proposed Reverse Converter
Area (um^2)	2253	648
Delay (ns)	2.099	1.565
Power (nW)	0.261	0.0518
Area-delay product	4729.047	1014.12
Power-Delay product	0.547	0.081

Table2

Constraints	Proposed Reverse Converter	Proposed Forward Converter
Area (um^2)	648	635
Delay (ns)	1.565	0.889
Power nW)	0.0518	0.048
Area-Delay Product	1014.12	564.515
Power-Delay Product	0.081	0.042

Table 3

Setup mode	ALL	In2Out
WNS	0.017	0.017
TNS	0.000	0.000
Violating Paths	0	0
All Paths	20	20



Specifications of the Chip:

Total Area: 21326.261 um^2

Chip density: 37.762 %

Standard Cells: 551

Total Nets: 618

Fig. 6 Chip Layout

VI. Conclusion

In this paper two different approaches for residue number system (RNS) representation were proposed. First, a reverse converter for RNS to conventional binary representation. In this approach a balanced five moduli set is vertically extended to improve the parallelism and dynamic range of the design.

Synthesis results suggest that the proposed vertical extension allows improving the conversion efficiency or area-delay product by up to 4.67 times. Secondly, a conventional forward converter was proposed for binary to RNS representation. Synthesis results also suggest that the conventional binary to RNS representation is more area, delay and power efficient than the complex RNS reverse conversion. Simulation and synthesis of the design is done successfully using the Cadence NCLaunch and RTL compiler respectively.

Placement and routing of the design is done at 45nmTsmc technology library. The timing reports at each and every step of design are observed and if any violations occurred they are removed using optimization technique. The density of the layout after detailed routing is finally reported as 100%.

References

[1] Hector Pettenghi and Leonel Sousa. "RNS Reverse Converters for Moduli Sets with Dynamic Ranges up to *(8n+1)."*Member, IEEE, Ricardo Chaves, Member, IEEE, Senior v Member, IEEE

[2] A. Hiasat, "VLSI implementation of new arithmetic residue to binary decoders," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 1, pp. 153–158, Jan. 2005.

[3] P. Mohan, "RNS-to-binary converter for a new three-moduli set," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 9, pp. 775–779, Sep. 2007.
[4] Hariri,K.Navi, and R. Rastegar, "A newhigh dynamic range moduli set with efficient reverse converter," *Trans. Comput. Math. Appl.*, pp. 660–668, Apr. 2008.
[5] Omar Abdelfattah, "Data Conversion in Residue Number System." Department of Electrical Computer Engineering McGill University Montreal, Canada January 2011.
[6] A. Skavantzos and M. Abdallah, "Implementation issues of the two level residue number system with pairs of conjugate moduli," *IEEE Trans. Signal Process.* vol. 47, no. 3, pp. 826–838, Mar. 1999.
[7] A. Skavantzos and Y. Wang, "New efficient RNS-to-weighted decoders for conjugate-pair moduli residue number systems," in *Proc.33rd Asilomar Conf. Signals, Systems, Comput.*, 1999, vol. 2, pp. 1345–1350.
[8] A. Skavantzos, "An efficient residue to weighted converter for a new residue number system," in proc. Great Lakes Symp. VLSI, 1998, pp. 185-191.
[9] J. Sorenson, "Two fast GCD algorithms," Trans.J.Algor., vol. 16, no. 1, pp. 110-144, Jan. 1994.
[10] H. Pettenghi, R. Chaves, and L. Sousa, "Multiplicative Inverses for Moduli Sets with Dynamic Ranges of 5n+β-bit," INESC-ID Tec. Rep., no. 15/2012, May 2012 [Online]. Available: http://sips.inesc-id.pt/~hector/RNS/TR15.pdf
[11] H. Vergos, D. Bakalis, and C. Efstathiou, "Efficient modulo 2n + 1 multi-operand adders," in Proc. IEEE 15th Int. Conf. Electron., Circuit Syst., 2008, pp. 694-697.
[12] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electronic Computers*, vol. EC-13, pp. 14–17, Jun. 1964.
[13] A. Omondi and B. Premkumar, *Residue Number Systems: Theory and Implementation*. London, U.K.: Imperial College Press, 2007.
[14] Z. Wang, G. Jullien, and W. Miller., "An efficient tree architecture for modulo 2n + 1 multiplication," *Trans. J. VLSI Signal Processi.*, vol. 14, pp. 241–248, 1996.
[15] R. Zimmermann, "Efficient VLSI implementation of modulo (2n + 1) or (2n - 1) addition and multiplication," in *Proc. IEEE 14th Symp. Comput. Arithmetic*, 1999, pp. 158–167.