ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

A Partial Residue to Decimal Converter for the Moduli Set $\{2^n - 1, 2^n, 2^n - 1\}$

Alhassan Abdul-Barik

Senior Lecturer, Department of Computer Science, Faculty of Mathematical Sciences, University for Development Studies, Tamale, Ghana

ABSTRACT: Residue Number System (RNS) is a non-conventional number system that uses remainders to represent numbers. These remainders, called residues are converted back to decimal representations using two types of converters. A forward converter converts from decimal representation to residue whiles a reverse converter converts from residues to decimal representations respectively. In this paper, a fast residue-to-decimal converter for the moduli set $(2^n - 1,\ 2^n, 2^n + 1)$ is proposed. The algorithm used to convert the residues of the moduli to decimal number is based on the Chinese Remainder Theorem (CRT). The approach does not allow full reverse conversion. The resulting implementation is based on two carry-save adders, two carry-propagate adders and a multiplexer. After comparing the proposed scheme to the state-of-the-art reverse converters in terms of area and delays, it is considered to be faster.

KEYWORDS: Residue Number System; Converter; Adder; Chinese Remainder Theorem

I. INTRODUCTION

Numbers play an important role in computer systems. Numbers are the basis and object of computer operations. The main task of computers is computing, which deals with numbers all the time [7]. For that matter, the study of number systems is very necessary as computers generally work on numbers. Residue Number Systems (RNS) is a non-weighted number system that utilizes remainders to represent numbers. In RNS, a set of moduli are chosen which are independent of each other. An integer is represented by the residue of each modulus and the arithmetic operations are based on the residues individually. Let $\{m_1, m_2, \ldots, m_n\}$ be a set of positive integers all greater than 1. $m_i$is called a modulus, and the n-tuple set $\{m_1, m_2, \ldots, m_n\}$ is called moduli set. Consider an integer number X. For each modulus in $\{m_1, m_2, \ldots, m_n\}$, we have $x_i = \mathrm{X}\, mod \mathrm{m_i}$(denoted as $|X|_{m_i}$). Thus, a number X in RNS can be represented as $X = \{x_1, x_2, \ldots, x_n\}$ [7].

Now, to calculate the number X from its residues, the Chinese Remainder Theorem (CRT) is formulated and applied as follows:

$$\mathrm{X} = \left|\sum_{i=1}^{n} \mathrm{M_i} \left|\mathrm{M_i}^{-1}\right|_{\mathrm{m_i}} \mathrm{x_i}\right|_{\mathrm{M}} \quad \textbf{(1)}$$

where; $\mathrm{M} = \prod_{i=1}^{N} \mathrm{m_i}\,; \mathrm{M_i} = \frac{\mathrm{M}}{\mathrm{m_i}}; \left|\mathrm{M_i} \times \mathrm{M_i}^{-1}\right|_{\mathrm{m_i}} = 1$

The CRT provides an algorithmic solution of decoding the residue encoded number back into its conventional representation. This theorem is considered the cornerstone in realizing the utilization of RNS [2]. Residue Number System (RNS) belongs to the second school of thought, to enhance system performance, it was proposed for computation-intensive application design because of its ability to support high-speed concurrent arithmetic [8]. These RNS features has been put to good use in various Digital Signal Processing (DSP) applications [1]. Today RNS is also regarded as one of the most popular techniques for reducing the power dissipation and the computation load in Very Large Scale Integrated Circuits (VLSI) system design [9].

RNS is a non-weighed number system with special carry characteristics and a potential that results in high computations. In RNS, addition, subtraction and multiplication are inherently carry-free, for instance, each digit of the result is a function of only one digit from each operand, hence independent of all other digits. As a result of the carry-free property, it is feasible to mechanize operations such as addition, subtraction and multiplication. These inherent

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

features enable RNS utilization in the fields of Digital Signal Processing (DSP), computational intensive areas such as; digital filtering, convolutions, correlations, Discrete Fourier Transform (DFT) computations, Fast Fourier Transform (FFT) computations and direct digital frequency synthesis [6], [5].

However, irrespective of the fact that, the residue number system supports high-speed parallel arithmetic, it is not very popular in processor construction due to the following difficult RNS arithmetic operations: reverse conversion between residue and binary numbers, overflow detection, magnitude comparison, sign detection, moduli selection, etc. However, solutions to some of these problems are currently being developed as a result of ongoing research. Out of these numerous RNS challenges, moduli selection and reverse conversion are the two most critical issues [4]. Many interesting reverse converters have been proposed for many moduli sets such as $(2^n - 1, 2^n, 2^n + 1)$ [10], $(2^n - 1, 2^n, 2^{2n+1} - 1)$ [3], to mention just a few. In [10] a new and uniform Adder Based algorithm using the New Chinese Remainder Theorem (CRT) for the RNS to binary conversion is presented whilst [3] presents a new residue to binary converter based on Mixed-Radix Conversion (MRC).

In this paper, the proposed scheme uses partial reverse conversion based on the CRT technique.

II. RELATED WORK

In a study by Molahosseini and Navi (2007), two efficient residue to binary converters for the new three-moduli set $\{2^n, 2^{n+1} + 1, 2^{n+1} - 1\}$ is presented. The proposed moduli set consists of pairwise relatively prime and balanced moduli, which can offer fast internal RNS processing and efficient implementation of the residue to binary converter. The new Chinese Remainder Theorem (CRT-I) is applied to derive an efficient residue to binary conversion algorithm for the new three-moduli set. Hardware implementation of the proposed residue to binary converters for the moduli set consist of one $(2n+2)$-bit carry save adder (CSA) with End Around Carry (EAC) and a modulo $(2^{2n+2}-1)$ adder. The proposed residue to binary converters are memoryless. In comparison with other residue to binary converter for a three-moduli set, the proposed converters have better area-time complexity whiles Hiasat and Abdel-Aty-Zohdy (1995), introduced a new algorithm for implementing residue to binary conversion for the moduli set $(2^k, 2^k-1, 2^{k-1}-1)$. The algorithm incorporates new compact forms for the multiplicative inverses. Binary adders are used in this proposed algorithm for the hardware implementation. If the system is pipelined, then the throughput rate of the converter increase to the equivalent of that of a single (3k-1) bits binary adder. Thus, hardware requirements and execution time are less, with corresponding larger dynamic range. Therefore, this moduli set could have an increasing role in designing residue-based arithmetic units for different computing applications. In 2011, Stamenkovic and Jovanovic published an alternative architecture derived by Mixed-Radix Conversion (MRC) for a four-moduli set. Due to the use of simple multiplicative inverses of the proposed moduli set, there is considerably reduction in the complexity of the RNS to binary converter based on the MRC. The hardware architecture for the proposed converter is based on the adders and subtracters, without the needed Read Only Memory (ROM) or multipliers. The implementation consists of two levels. The first level, is the algorithm to convert RNS number to mixed-radix digits. The algorithm is improved by using optimal choice of form of moduli set. The second level is a simplified hardware architecture. Carry-Save-Adder (CSA) with End-Around-Carry (EAC) is replaced with Borrow-Save-Subtracter that avoids two complement operations, and EAC adder. Further, the binary subtraction is optimized by using Borrow-Propagate-Subtracter (BSS) with End-Around-Borrow (EAB) which avoids one complement operation and the use of multiplexer(s). The proposed converter architecture is memoryless enabling efficient implementation. However, Hariri, Rastegar, and Navi (2006) studied the moduli set $\{2^n, 2^{2n}-1, 2^{2n}+1\}$ and proposed a reverse converter. This moduli set provides the dynamic range (DR) of $2^n \times (2^{4n}-1)$ and the implementation results have shown that its reverse converter has better area and time complexities in comparison with the moduli sets with the same dynamic range categories. The authors showed that for majority of the similar dynamic ranges, the proposed reverse converter is faster than the reverse converter of $\{2^n-1, 2^n, 2^n+1\}$ but the reverse converter of $\{2^n-1, 2^n, 2^n+1\}$ has less area.Pettenghi et al. (2013),also proposed methods to design memoryless reverse converters for the proposed moduli sets with large dynamic ranges, up to (8n+1)-bit. Due to the complexity of the reverse conversion, both the Chinese Remainder Theorem (CRT) and the Mixed Radix Conversion (MRC) are applied in the proposed methods to derive efficient reverse converters. Experimental results suggest that the proposed vertical extensions allows the reduction of the area-delay-product up to 1.34 times in comparison with the related state-of-the-art. The horizontal extensions allow larger and more balanced moduli sets, resulting in an improvement of the RNS arithmetic computation, at the cost of lower reverse conversion performance.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

III. PROPOSED ALGORITHM

Given the RNS number $X = (x_1, x_2, x_3)$ with respective moduli set $\{2^n - 1, 2^n, 2^n + 1\}$, where; $m_1 = 2^n - 1, m_2 = 2^n, \text{and} m_3 = 2^n + 1$, then the following holds true;

$M_1 = 2^n(2^n + 1)$; $M_2 = (2^{2n} - 1)$; and $M_3 = 2^n(2^n - 1)$ **(2)**

Theorem 1*:* For the given moduli set, then;

$|M_1^{-1}|_{m_1} = |2^{n-1}|_{m_1}$ **(3)**

$|M_2^{-1}|_{m_2} = |-1|_{m_2}$ **(4)**

$|M_3^{-1}|_{m_3} = |-2^{n-1}|_{m_3}$ **(5)**

The proof of **(3)** – **(5)** is demontrated in [5].

Theorem 2: For the given moduli set, any RNS number X can be represented as;

$$X = 2^n\xi + x_2 \quad \textbf{(6)}$$

where;

$$\xi = \left|\left|\begin{matrix}|(2^n x_1 + x_1)2^{n-1}|_{2^{2n}-1} + |-2^n x_2|_{2^{2n}-1} \\ +|-x_3|_{2^{2n}-1} + |+2^{n-1}x_3|_{2^{2n}-1}\end{matrix}\right|_{2^{2n}-1}\right| \quad \textbf{(7)}$$

Proof: Substituting equations **(2)** through to **(5)** into **(1)** and factorizing out 2^nwe obtain **(6)**.

HARDWARE IMPLEMENTATION

Here, the mathematical formulations and simplifications for obtaining an effective and robust design is presented. The design and usage of simplified architecture that is cost effective and less complex is also presented.

A. *Design Considerations and Mathematical Simplifications*

Hardware implementation considers significantly, the minimisation of production cost and cost of usage. It is therefore important to further simplify equation (7) to equation (8) to reduce the implementation cost, which is achieved by reducing equation (7) to utilise Adders and Multiplexers.

Equation (7) can further be simplified as follows;

$\xi = |\varphi_1 + \varphi_2 + \varphi_3 + \varphi_4|_{2^{2n}-1}$ **(8)**

where;

$\varphi_1 = |(2^n x_1 + x_1)2^{n-1}|_{2^{2n}-1}$ **(9)**

$\varphi_2 = |-2^n x_2|_{2^{2n}-1}$ **(10)**

$\varphi_3 = |-x_3|_{2^{2n}-1}$ **(11)**

$\varphi_4 = |2^{n-1}x_3|_{2^{2n}-1}$ **(12)**

Now, considering equations (9)-(12) and simplify them for implementation in a VLSI system. It is necessary to note that $x_{i,j}$ means the j-th bit of x_i.

Evaluation of $\boldsymbol{\varphi_1}$

The residue x_1 can be represented as follows;

$x_1 = x_{1,n-1} \dots x_{1,1}x_{1,0}$ **(13)**

Thus,

$$|(2^n x_1 + x_1)2^{n-1}|_{2^{2n}-1} =$$

$$\left|2^{n-1}\left(\underbrace{x_{1,n-1}\dots x_{1,0}\overbrace{0\dots0}^{n\ Bits}}_{2n-bits} + \underbrace{\overbrace{0\dots0}^{n\ Bits}x_{1,n-1}\dots x_{1,0}}_{2n}\right)\right|_{2^{2n}-1}$$

$$= \left|2^{n-1}\left(\underbrace{x_{1,n-1}\dots x_{1,1}x_{1,0}x_{1,n-1}\dots x_{1,1}x_{1,0}}_{2n-bits}\right)\right|_{2^{2n}-1}$$

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

$$= \overbrace{\overline{x}_{1,0}x_{1,n-1}\cdots x_{1,1}x_{1,0}}^{n+1\ Bits}\underbrace{x_{1,n-1}\cdots x_{1,n+2}x_{1,1}}_{n-1} \quad \textbf{(14)}$$

Evaluation of φ_2:

The residue x_2 can be represented as follows;

$$x_2 = x_{2,n-1}\cdots x_{2,1}x_{2,0} \quad \textbf{(15)}$$

Therefore,

$$|-2^n x_2|_{2^{2n}-1} = \overline{x}_{2,n-1}\cdots\overline{x}_{2,1}\overline{x}_{2,0}\overbrace{11\ldots11}^{n\ Bits} \quad \textbf{(16)}$$

Evaluation of $\varphi_3$and φ_4:

The residue x_3 can be represented as follows;

$$x_3 = x_{3,n}\cdots x_{3,1}x_{3,0} \quad \textbf{(17)}$$

Therefore,

$$\varphi_3 = |-x_3|_{2^{2n}-1} = \overbrace{11\ldots11}^{n\ Bits}\underbrace{\overline{x}_{3,n-1}\cdots\overline{x}_{3,1}\overline{x}_{3,0}}_{n\ Bits} \quad \textbf{(18)}$$

Again,

$$\varphi_4 = |2^{n-1}x_3|_{2^{2n}-1} = \underbrace{0x_{3,n}\cdots x_{3,1}x_{3,0}}_{n+1\ Bits}\overbrace{00\ldots00}^{n-1\ Bits} \quad \textbf{(19)}$$

B. *The Proposed Architecture*

$\xi$is computed according to equation (8) where all the parameters are defined in equations (9) – (12). Carry Save Adders (CSAs) and Carry Propagate Adders (CPAs) are used to reduce the hardware complexity. As shown in Figure 1, ξ is computed using CSAs 1and 2 and two regular $2n$-bit CPAs 1 and 2. The results of these CPAs are passed on to a multiplexer (MUX 1) which would then pass either of them down. MUX 1 will pass on the result of CPA 1 if the carry out of CSA 1 is a '0', otherwise the result of CPA 2 is passed on.

CSAs 1 and 2 require an area of $2n\Delta_{FA}$ each as well as CPAs 1 and 2. Therefore, in order to obtain ξ, a total area of $8n\Delta_{FA}$ will be required.

Regarding the delay, each CSA (i.e. CSAs 1 and 2) impose a delay of D_{FA} while the CPA pair 1 and 2 impose a delay of $2nD_{FA}$ since they are in parallel, thus delay imposed on computing $\xi$is $(2n + 2)D_{FA}$. The final computation for (6) is a concatenation which does require any hardware.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

The schematic diagram for the proposed scheme is shown below;

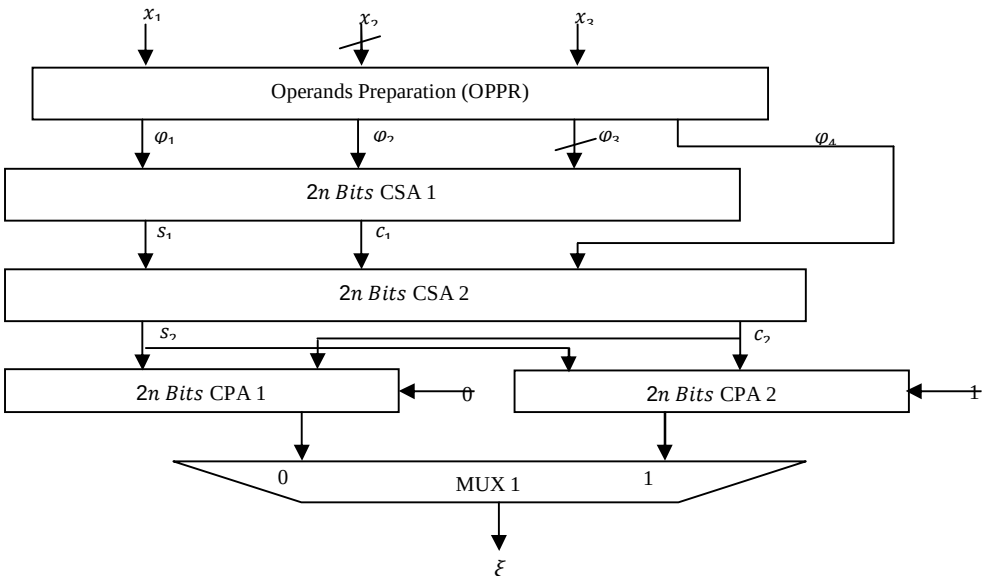

Figure 1: Block Diagram of the Reverse Converter

IV. SIMULATION RESULTS

In this section, the performance of the proposed system is evaluated by comparing it with [10] and [3]. The evaluation is done in terms of area (A), delay (D) and AD^2. From the AD^2 comparison, it can be concluded that, the proposed scheme is better.

Table 1: Area, Delay, AD^2 Comparison

Converters	Area	Delay	AD^2
[10]	4n	4n+2	$64n^3+64n^2+16n$
[3]	9n+2	10n+5	$900n^3+1100n^2+425n+50$
Proposed Scheme	8n	2n+2	$32n^3+64n^2+32n$

The graph below in Figure 2 shows the performance of the various schemes. In the graph, it is clear that the proposed scheme is more efficient in terms of area and delay comparison.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 11, November 2018

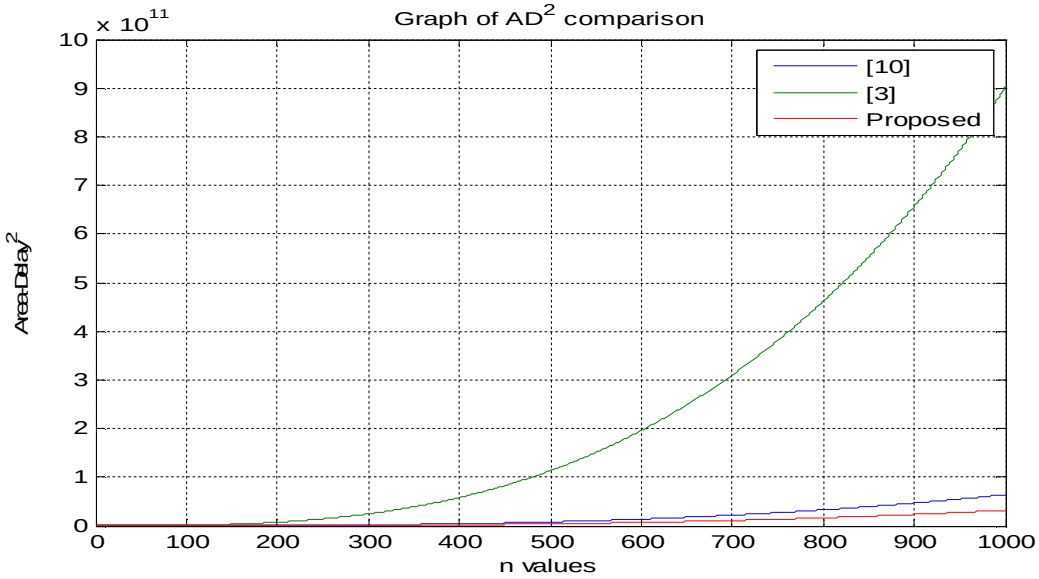

Figure 2: Graph of Area and Delay Comparison

V. CONCLUSION AND FUTURE WORK

In this paper, a fast residue-to-binary converter for the moduli set $(2^n - 1, 2^n, 2^n + 1)$ has been proposed. The approach does not require full RNS-binary conversion. Theoretical analysis from Table 1 shows that, the proposed scheme is more efficient as compared to the state-of-art schemes in [3] and [10] in the Area-Delay comparison

REFERENCES

1. Nannarelli, A., Re, M., and Cardarilli, G. C., "Tradeoffs between Residue Number System and Traditional FIR Filters", IEEE Int. Symp. Circuits Syst. pp. 305–308, 2001.
2. Omondi, A., and Premkumar, A., "Residue Number System Theory and Implementation (Vol. 2). Imperial College Press, 2007.
3. Molahosseini, A. S., Navi, K., and Rafsanjani, M. K. "A New Residue to Binary Converter based on Mixed-Radix Conversion," 3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA 2008), pp. 1-6, 2008.
4. Cao, B., Chang, C. H, and Srikanthan, T., "An Efficient Reverse Converter for the 4-Moduli Set $(2^n - 1, 2^n, 2^n + 1, 2^{2n} - 1)$," IEEE Trans. on Circuits and Systems, vol 50, pp. 1296-1303, 2003.
5. Bankas, E. K., and Gbolagade. K. A. "A New Efficient FPGA Design of Residue-to-Binary Converter". International Journal of VLSI design & Communication Systems (VLSICS) Vol.4, No.6, 2013.
6. Gbolagade, K. A., "An Efficient MRC based RNS-to Binary Converter for the $\{2^{2n}-1, 2^n, 2^{2n+1}-1\}$ Moduli Set". International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 4, 2013.
7. Lu, M., "Arithmetic and Logic in Computer Systems", John Wiley & Sons, Inc., Hoboken, New Jersey 2004.
8. Sheu, M., Lin, S. H., Chen, C., and Yang, S. W. "An efficient VLSI Design for a Residue to Binary Converter for General Balance Moduli Set $\{2^n-3, 2^n-1, 2^n+1, 2^n+3\}$", IEEE Transactions on Circuits and Systems II, 51(3), pp. 152–155, 2004.
9. Stouratitis, T. and Paliouras, V. "Considering the Alternatives in Lowpower Design," IEEE Circuits and Devices, pp. 23–29, 2001.
10. Wang, Y., Song, Y., Aboulhamid, M., and Shen, H., "Adder Based Residue to Binary Number Converters for $(2^n - 1, 2^n, 2^n + 1)$", IEEE Trans. on Signal Processing, vol. 50, pp. 1772-1779, 2002.
11. Hariri, A., Rastegar, R., & Navi, K, "High Dynamic Range 3-Moduli Set with Efficient Reverse Converter", 2006.
12. Hiasat, A. A., and Abdel-Aty-Zohdy, H. S., "An Algorithm for the Design of A Residue-To-Binary Converter", Midwest Symposium on Circuits and Systems, 1280-1283, 1995.
13. Pettenghi, H., Chaves , R., and Sousa, L., "Residue Number System Reverse Converters for Moduli Sets with Dynamic Ranges up to (8n+1)-bit", IEEE, 1487-1500, 2013.
14. Stamenkovic, N., & Jovanovic, B., "Reverse Convertor Design for the 4-Moduli Set {2^n-1,2^n+1,2^(2n+1)-1} Based on the Mixed-Radix Conversion", SER.: ELEC.ENERG, 89-103, 2011.