iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 Issue: IV Month of publication: April 2018

DOI: http://doi.org/10.22214/ijraset.2018.4643

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue IV, April 2018- Available at www.ijraset.com

VLSI Implementation of a Reverse Converter for a Class of Six Moduli Set

M. Hemalatha[1], J. Sangavi[2], M. Anbarasi[3]
[1, 2, 3,] PG Scholars Dept of Electronics and Communication Engineering, Pondicherry Engineering College.

Abstract: The application of RNS to digital signal processing is the ability to operate on signed numbers. This has led to the development of the alternative number systems such as residue number system (RNS). Reverse converter design (Residue to binary) is a time consuming process and thus the design of efficient reverse converter is highly important. This increases the chip area and also increases the power consumption. To overcome the above drawback, to design an efficient reverse converter for a class of 6-moduli set to increase the dynamic range. The idea behind this approach is that modular adders have been designed targeting the RNS reverse converter, which require modulo adders of 16 bit width and to obtain the output of the reverse converters. However, the RCA-based design is more energy efficient than the prefix based.
Index Terms: Computer arithmetic, modulo adder, residue arithmetic, residue number system (RNS), ripple carry adder (RCA).

I. INTRODUCTION

Power consumption and performance of today's system gains great importance as technology gets scaled down. Parallelism can be exploited at different levels such as algorithm, architecture and at the system level. Although significant improvements have been made at the architectural level, the majority of today's processors still suffer from the drawbacks due to the conventional weighted binary number system. This has led to the development of the alternative systems such as residue number system (RNS). Residue number system (RNS) operations have no carry between different residue digits [11], [12]. Addition, subtraction, and multiplication run on a residue digit independently from computations on other digits [11], [12]. This independency reduces the time needed to perform residue-based arithmetic operations compared with binary based operations. Therefore, RNS is used efficiently in some digital signal. Processing and cryptographic applications that require high speed computations [12]–[16]. It is a carry-free arithmetic with modular characteristics that offers the potential for high-speed and parallel computation. Arithmetic operations, such as addition, subtraction, and multiplication, can be carried out independently. The adoption of the RNS has provided significant efficiency for different types of Digital Signal Processing (DSP) applications [17], such as filtering [18], computation of the Discrete Fourier Cosine transform [19], communication [20], and cryptography [21], [22]. 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 applications [23], [24]. 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. Reverse converter design (Residue to binary) is a time consuming process and thus the design of efficient reverse converter is highly important. To overcome the drawback of five moduli set (i.e,) increase the chip area and also increase the power consumption, to design an efficient signed reverse converter for a class of 6-moduli set to increase the dynamic range. The idea behind this approach is that modular adders have been designed targeting the RNS reverse converter, which require modulo adders of different sizes and to obtain signed representations at the output of the reverse converters. However, the RCA-based design is more energy efficient than the prefix based. The algorithms 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) [25]. RNS systems that can import negative numbers through forward converters and operate on them, while reverse converters restrict the representation of the results to positive numbers. This design based on a general conversion procedure uses several lookup tables (LUTs) in order to achieve a suitable implementation for field-programmable gate arrays (FPGA). However, LUT size grows exponentially with the increasing the bit width.

The most important considerations when designing RNS Systems is the choice of the moduli set. The choice of moduli affects the complexity of forward and reverse converters as well as RNS arithmetic circuits. In (A systematic approach for selecting practical moduli sets for residue number systems), Abdallah and Skavantzos state that the moduli set, S = $m_1,\dots, m_L$, should be chosen such that the moduli m_i satisfy the following criteria:

A. They should be pairwise prime. That is, gcd (m_i, m_j) = 1 for all m_i = m_j.
B. Each moduli m^i should be as small as possible so that operations modulo mi require minimum computational time.
C. The moduli m^i's should imply simple binary to RNS and RNS to binary conversions as well as simple RNS arithmetic.
D. The moduli product should be large enough to implement the desired dynamic range.
E. The moduli should provide a well balanced decomposition of the dynamic range. This means that the difference in word length between the moduli should be as small as possible.

A new approach was recently proposed to improve the efficiency of unsigned reverse converters by adopting specialized adder components [2]. The idea behind this approach is that modular adders have been designed targeting the individual RNS arithmetic channels and not for implementing reverse converters, which require modulo adders of 16 bit width. This observation has motivated the work presented herein, which develops specialized adder components to obtain signed representations at the output of the reverse converters. In particular, this paper presents adder components that can be easily used in reverse converter structures for a class of moduli sets. The moduli sets with {2^n, 2^n-1, 2^n+1, 2^n-$2^{(n+1)/2}$+1, 2^n+$2^{(n+1)/2}$+1, 2^{n-1}+1}, (8n+1)-bit DR are proposed by extending the original moduli set. The extensions results in the moduli sets {$2^{n+\beta}$, 2^n-1, 2^n+1, 2^n-$2^{(n+1)/2}$+1, 2^n+$2^{(n+1)/2}$+1, 2^{n-1}+1}with DRs up to (8n+1)-bit when β = 2n and 2n+1+1 is used. However, the design of residue-based arithmetic components for the moduli of the form (2^n±$2^{((n+1)/2)}$+1) is more demanding compared with moduli of the form (2^n ±1) [26], [27].

II. RESIDUE NUMBER SYSTEM

A. RNS Unsigned and Signed Integers

The first step to set up an RNS is the definition of a moduli set. A moduli set is composed of pair wise relatively prime numbers {m_1,m_2,...,m_n}, which define a dynamic range of M numbers with a single and unique representation; i.e., the set of residues {R_i ≡ X mod m_i|1 ≤ i ≤ n} uniquely identify a congruence class modulo M, where M = m_1 × m_2×⋯×m_n. Therefore, this range can be used to represent unsigned or signed numbers. While unsigned numbers with a magnitude within the range of 0 to M−1 are represented by the remainders of their magnitude; with signed numbers the dynamic range is split into half, and one half is reserved for positive numbers, whereas the other represents negative numbers.

$$\text{for M even:} \begin{cases} [0, \frac{M}{2} - 1] \cap N: positive \\ [\frac{M}{2}, M - 1] \cap N: negative \end{cases} \quad (1)$$

$$\text{for M odd:} \begin{cases} [0, \frac{M-1}{2}] \cap N: positive \\ [\frac{M+1}{2}, M - 1] \cap N: negative \end{cases} \quad (2)$$

B. Design of Reverse Converter

The reverse converter is basically a residue to binary converter. It is the important part of residue number system because the speed efficiency obtained in performing calculation in residue domain should not be degraded while converting it to the binary number system. Hence the design of speed efficient reverse converter has significant that without speed efficacy, the use of residue number system that is speed and power efficient cannot be used in processors, therefore it is necessary to have a speed reverse conversion process. For a speed reverse conversion process there are three main steps to be carried:
1) Judicious selection of the moduli set
2) A dynamic range suitable for the application
3) A conversion algorithm compatible with the properties of the selected moduli set

C. Considered Moduli Set

A large number of moduli sets have been proposed for RNS. Some larger moduli sets, such as {2^n−1,2^n,2^n+1,2^{n+1}−1,$2^{n−1}$−1}[1] and {2^n, 2^n-1, 2^n+1, 2^n-$2^{(n+1)/2}$+1, 2^n+$2^{(n+1)/2}$+1, 2^{n-1}+1}, allow more balanced and efficient arithmetic channels at the cost of more complex reverse converters. On the other hand, other moduli sets are defined to achieve simpler reverse converter structures due to the mathematical relations between the moduli. This latter class of moduli sets, which also leads in general to simple arithmetic channels in the channels (for a near power of two values), have been applied to develop RNS-based DSPs [26].

This is exactly the class of moduli sets targeted in this paper, which are referred from now on as c-class moduli sets. Note that the selection of the moduli set depends on the targeted applications, number of arithmetic operations, and dynamic range, among other specifications.The moduli sets $\{2^n+2^n-1, 2^n+1, 2^n-2^{(n+1)/2}+1, 2^n+2^{(n+1)/2}+1, 2^{n-1}+1\}$have been proposed for RNS. An interesting common feature of these sets is the fact that one modulo has the form 2^k while the product of the other moduli takes the value of $2^P -1$. Therefore, such moduli sets can have a common template in the form of $\{2^k, 2^P-1\}$, which is herein designated composite moduli set. The Chinese remainder theorem (CRT) and its extension, i.e., the New CRT-I, have been used up to now for designing reverse converters for the c-class moduli sets. By CRT, an RNS number can be converted into the weighted number X as follows:

$$X = \sum_{i=1}^{n} \left| X_i N_{iPi} M_i \right|_M$$

Where $M= P_1, P_2, P_3 \ldots P_n$, $M_i=M/P_i$ and $N_i=\left| Mi \right|^{-1}_{Pi}$is the multiplicative inverse of Mi modulus Pi. The CRT can be implemented in parallel channels followed by a modulus M adder. This modulus adder is very large and can be result in inefficient hardware implementation of the reverse converter. The CRT is used for deriving reverse converters for the six-moduli set $\{2^n+2^n-1, 2^n+1, 2^n-2^{(n+1)/2}+1, 2^n+2^{(n+1)/2}+1, 2^{n-1}+1\}$.

D. New Chinese Remainder Theorem (crt)

The weighted number X can be composed by New CRT-I as follows:

$$X=x_1+P^*\left| k_1 (x_2-x_1) + k_2P_2 (x_3-x_2)+\ldots+k_{n-1} P_2 P_3\ldots P_{n-1} (x_n-x_{n-1})\right|_{P_2P_3P_n} \quad (3)$$

Where, k-multiplication

P-moduli set element

$$\left| k_1 \times P_1 \right|_{P_2P_3\ldots P_n} =1 \quad (4)$$

$$\left| k_2 \times P_1 \times P_2 \right|_{P_2P_3\ldots P_n} =1 \quad (5)$$

$$\left| k_{n-1} \times P_1 \times P_2 \ldots \times P_{n-1} \right|_{P_n}=1 \quad (6)$$

By New CRT-I, the size of the final modulo adder is reduced in comparison to the traditional CRT. In particular, if the first modulus of the moduli set is selected in the form 2^k, and the multiplication of the other moduli set is in the form 2^k-1,the New CRT-I can be implemented by only a multi-operand modulus adder. The reverse conversion formula for c-class moduli sets using the New CRT-I takes the form

$$X = x_1+2^kY \quad (7)$$

where x_1 corresponds to X mod 2^k and Y to $(X - x_1)2^{-k}$ mod $(2^P -1)$. Y is often computed using a tree of carry-save adders (CSAs) with end-around carry (EAC) followed by a carry-propagate adder (CPA) with EAC.

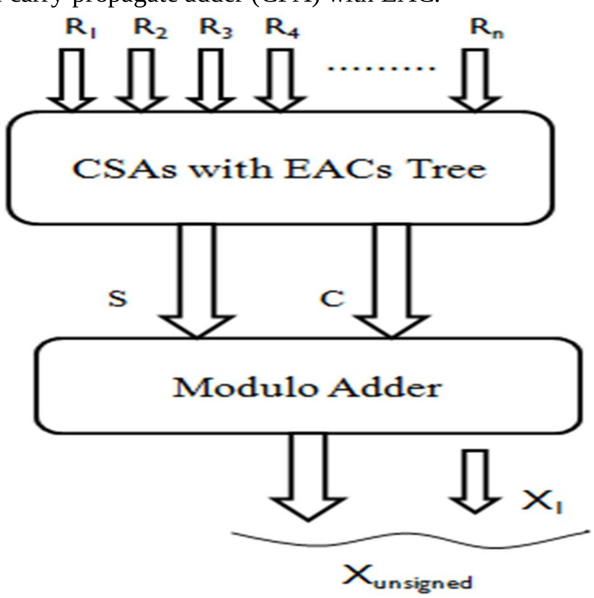


Fig. 1. Reverse converter structure for considered moduli sets.

In other words, the CSA tree produces S and C such that their addition corresponds to the result, as shown in Fig. 1. Then, these two binary vectors are added using an EAC modulo adder to produce the complete result, Y. Finally, the concatenation of the first residue, x_1, and Y produces the final unsigned result X, since x_1 is a k-bit number. The number of CSAs in the CSA-tree is dependent on the target moduli set. Due to the efficient characteristic of CSA, which does not propagate carries, since they are added on the subsequent stage, the structure presented in Fig. 1 is the most efficient existing reverse converter architecture for c-class moduli sets, namely for large dynamic ranges.

The logic operations involved in conventional carry select adder (CSLA) and binary to excess-1 converter (BEC)-based CSLA are analyzed to study the data dependence and to identify redundant logic operations as shown in Fig. 2. It consists of one HSG unit, one FSG unit, one CG unit, and one CS unit. The CG unit is composed of two CGs (CG0 andCG1) corresponding to input-carry'0'and'1'. The HSG receives two n-bit operands (A and B) and generate half-sum word s0 and half-carry word c_0 of width n bits each. Both CG0 and CG1 receive s_0 and c_0 from the HSG unit and generate two n-bit full-carry words c_{01} and c_{11} corresponding to input-carry '0' and '1', respectively. The CS unit selects one final carry word from the two carry words available at its input line using the control signal c_{in}. It selects c_{01} when c_{in} =0; otherwise, it selects c_{11}. The CS unit can be implemented using an n-bit 2-to-l MUX. However, find from the truth table of the CS unit that carry words c_{01} and c_{11} follow a specific bit pattern.

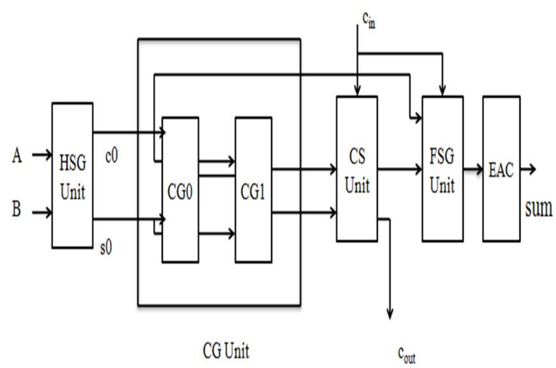


Fig. 2. Carry Select Adder design

Where,

HSG - Half-sum generator

$$S_0\ (i) = A\ (i) \oplus B\ (i) \tag{8}$$

HCG -Half-carry generator

$$C_0\ (i) = A\ (i)\ \text{AND}\ B\ (i) \tag{9}$$

FSG -Full-sum generator

$$S\ (0) = S_0\ (0) \oplus C_{in}\ S\ (i) = S_0\ (i) \oplus C\ (i\text{-}I) \tag{10}$$

FCG -Full-carry generator

$$C^0{}_1\ (i) = C_{01}\ (i\text{-}l)\ \text{AND}\ S_0\ (i)\ \text{OR}\ C_0\ (i)\ \text{for}\ (C^0{}_1\ (0) = 0) \tag{11}$$

$$C^1{}_1\ (i) = C_{11}\ (i\text{-}l)\ \text{AND}\ S_0\ (i)\ \text{OR}\ C_0\ (i)\ \text{for}\ (C^1{}_1\ (0) = 1) \tag{12}$$

$$C_{out} = C\ (n\text{-}l) \tag{13}$$

III. PROPOSED MODULAR ADDER COMPONENT

A new approach was recently proposed to improve the efficiency of unsigned reverse converters by adopting specialized adder components. The idea behind this approach is that modular adders have been designed targeting the individual RNS arithmetic channels and not for implementing reverse converters, which require modulo adders of different sizes and requirements. This observation has motivated the work presented herein, which develops specialized adder components to obtain signed representations at the output of the reverse converters.

In particular, adder components that can be easily used in reverse converter structures for a class of moduli sets, which is used to transform unsigned reverse converters to signed reverse converters, with a low overhead on performance, cost, and energy consumption. All the figures of merit, including chip area, delay, and power consumption, were improved with the proposed method in comparison with the traditional method.

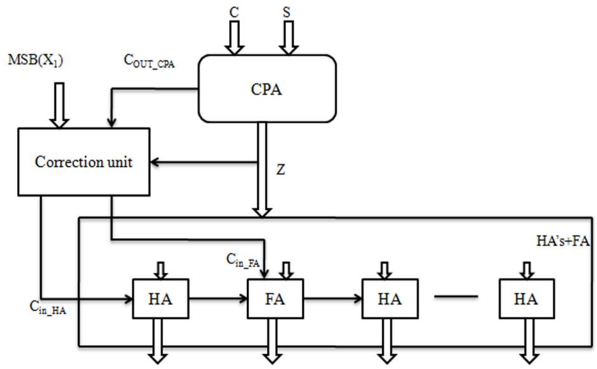


Fig. 3. RNS Modular Adder Component.

The output of the CPA used for the first addition stage should be increased only in two situations: when Cout_CPA =1 (corresponding to the EAC bit) or G_{one} = 1 (corresponding to the ones-detector output). If only one of these three signals (Sign, Cout_CPA, and G_{one}) takes the value 1, it is enough to have Cin_HA = 1 in Fig. 3. However, if the Sign signal takes the value 1 and simultaneously Cout_CPA or G_{one} is also 1 the value 2 has to be added to the result of the CPA. Thus the second HA is substituted into Fig. 3 by a full adder (FA) with the extra input Cin_FA. Table I shows the values of the Cin_HA and Cin_FA signals generated by the Correction Unit in Fig. 3 for all the different combinations of the signals Sign, Cout_CPA, andG_{one}. It should be noted that Cout_CPA and Gone are never both 1. Therefore,(14) and (15) can be applied to provide the required inputs to the HAs + FA unit in Fig. 2

$$C_{in\_HA} = Sign \oplus (G_{one} \vee C_{out\_CPA}) \quad (14)$$
$$C_{in\_FA} = Sign \wedge (G_{one} \vee C_{out\_CPA}). \quad (15)$$

The hardware structure of the proposed specialized adder component is shown in Fig. 3, where the HAs + FA unit performs all-ones, end-around carry, and negative number corrections controlled by the Correction Unit. The logic circuits of the Correction Unit, based on (15), are shown in Fig. 4.

A. Numerical Example

Consider the RNS reverse converter based on the six moduli set$\{2^n+2^n-1, 2^n+1, 2^n-2^{(n+1)/2}+1, 2^n+2^{(n+1)/2}+1, 2^{n-1}+1\}$ For (n>=5), the moduli set takes the values {32, 31, 33, 25, 41, 17} and the dynamic range is 352766400.

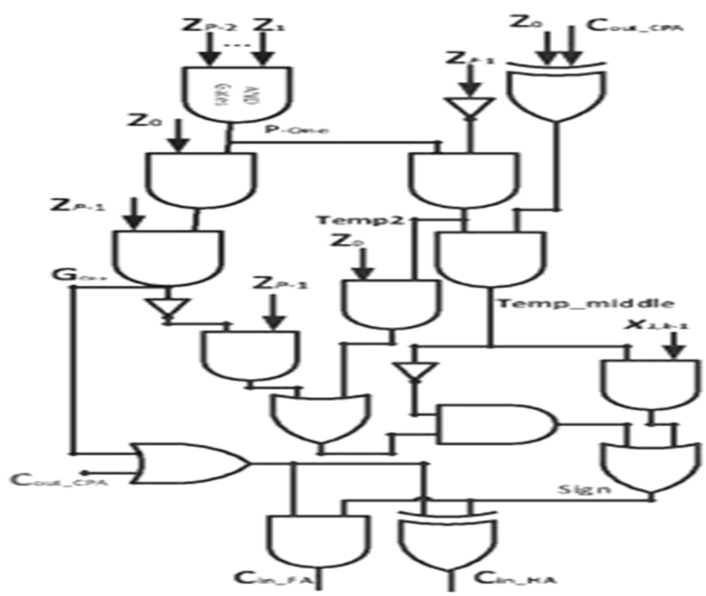
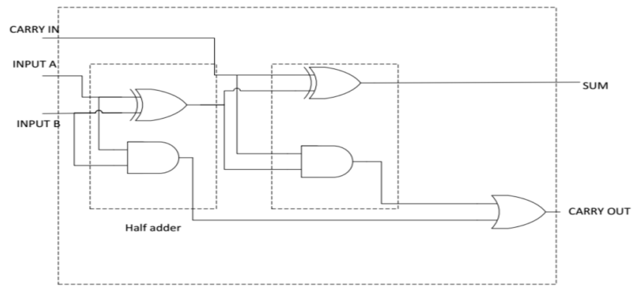


Fig. 4. Circuits for implementing the correction unit of Fig. 3



A full adder adds binary numbers and accounts for values carried in as well as out. A one-bit full adder adds three one-bit numbers, often written as A, B, and C_{in}; A and B are the operands, and C_{in} is a bit carried in from the previous less significant stage. The full-adder is usually a component in a cascade of adders, which add 8, 16, 32, etc. bit binary numbers. The circuit produces an unsigned two-bit output, output carry and sum typically represented by the signals C_{out} and S. A full adder can be implemented in many different ways such as with a custom transistor-level circuit or composed of other gates. The Boolean functions for the full adder in terms of exclusive-OR operations can be expressed as:

$$S = A \oplus B \oplus C_{in} \quad (16)$$

$$C_{out} = (A \text{ and } B) \text{ or } (C_{in} \text{ and } (A \oplus B)) \quad (17)$$

In this implementation, the final OR gate before the carry-out output may be replaced by an XOR gate without altering the resulting logic. The logic diagram for this multiple-level implementation consists of two half adders and an OR gate as shown in Fig. 6.

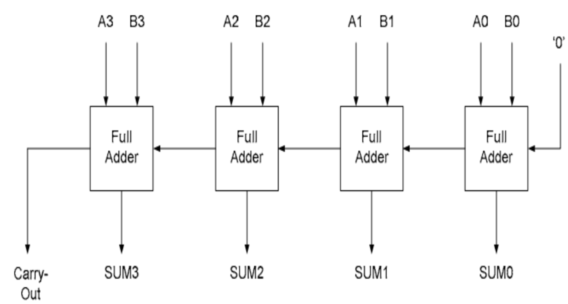


Fig.6. Logic Diagram for the Multiple-Level Implementation Consists of Two Half Adders and an OR Gate

IV. EXPERIMENTAL RESULTS

Experimental evaluation was performed for the proposed component alone and for the whole reverse converters for two different moduli sets from the c-class the five-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1, 2^{n-1}-1\}$ [5] and six moduli set $\{2^n, 2^n\text{-}1, 2^n+1, 2^n\text{-}2^{(n+1)/2}+1, 2^n+2^{(n+1)/2}+1, 2^{n-1}+1\}$. All the converters were described in synthesizable VHDL and verified using ModelSim. Using these HDL specifications, implementations were done targeting ASIC based on the TSMC 65-nm general-purpose standard cell library (TCBN65GPLUS, version 200A) tailored for the TSMC 65-nm CMOS logic. The experimental results for the chip area (μm^2), delay (ns), and power consumption (mW) are given in Tables I and comparison between fifth and sixth moduli set for area (μm^2), delay (ns), and power consumption (mW) are given in Table II.

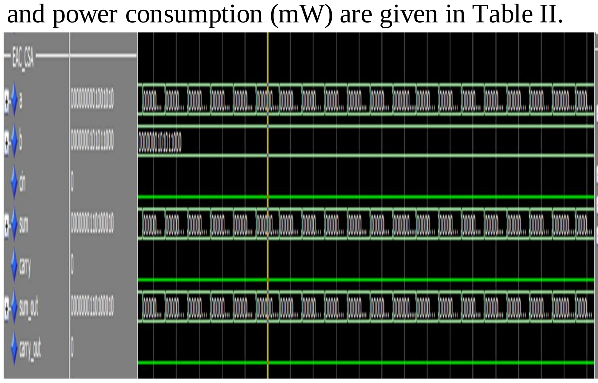


Fig. 7. Simulation results for CSA with EAC

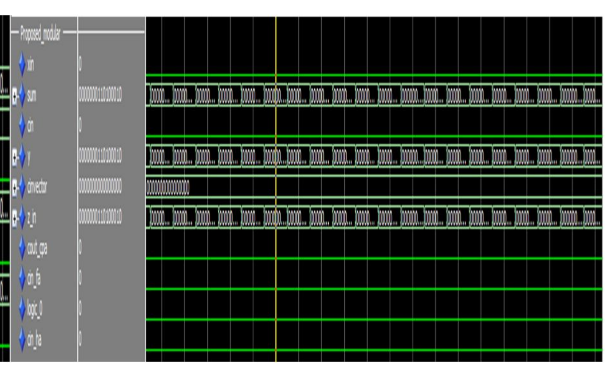


Fig. 8. Simulation Result for Proposed Modular Adder

In Fig. 7. shows the simulation results for CSA with EAC, two inputs and c_{in} is given for half sum generation, full sum generation, carry generation and carry select unit and the output of the carry select adder is given for end around carry which produces the sum=0000000110100010 and carry =0. In Fig. 8. shows the simulation result for Proposed modular adder. The output of the Carry save adder with end around carry is given as the input for carry propagate adder. The output of this is given to Correction unit and HA's which generates the output as 0000000110100010.

In this chapter, discuses the output of the RNS modular adder. This RNS modular consists of different functions namely carry save adder with end around carry and proposed modular adder. In proposed modular adder having carry propagate adder, correction unit and half adders and full adder.

The combined function of carry select adder and proposed modular adder gives the output for RNS modular adder.

Table i

Experimental result for the reverse converter of the six moduli set $\{2^n, 2^n-1, 2^n+1, 2^n - 2^{(n+1)/2}+1, 2^n + 2^{(n+1)/2}+1, 2^{n-1}+1\}$

Parameters	Delay (ns)	Power(mW)
SIX MODULI SET $\{2^n, 2^n-1, 2^n+1, 2^n - 2^{(n+1)/2}+1, 2^n + 2^{(n+1)/2}+1, 2^{n-1}+1\}$	35.385	64

Table ii

Experimental results for the fifth and sixth moduli set with modular adder

FIVE MODULI SET	Delay (ns)	36.41
	Power (mW)	80.9
SIX MODULI SET	Delay (ns)	35.385
	Power (mW)	64

V. CONCLUSION

In this paper, we presented a new specialized adder component to make reverse converter structures able to provide two's complement representation of negative RNS numbers. This adder component is flexible and can be easily applied to any reverse converter for the c-class moduli-sets, simply by replacing its final modulo adder. Residue-to-binary converters (reverse converters) has been designed for two different ranges using modular adder, carry select adder, end around carry and carry propagate adder. The performance of the reverse converters based on modular adder is analyzed. The proposed design outperforms the state of the art for achieving reverse converters by improving the delay, chip area, and energy consumption.

REFERENCES

[1] E. Zarandi, A. S. Molahosseini, M. Hosseinzadeh, and L. Sousa, "An Efficient component for designing signed reverse converters for the class of RNS Moduli Sets of composite form {2k,2P −1}," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, pp. 48 –59, issue. 1, March 2017.

[2] A. A. E. Zarandi, A. S. Molahosseini, M. Hosseinzadeh,S. Sorouri, S. Antão, and L. Sousa, "Reverse converter design via parallel-prefix adders: Novel components, methodology, and implementations," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 2, pp. 374–378, Feb. 2015.

[3] J. Chen and J. Hu, "Energy-efficient digital signal processing via voltage overscaling- based residue number system," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 7, pp. 1322–1332, Jul. 2013

[4] Z. Torabi and G. Jaberipur, "Low-Power/Cost RNS

[5] Comparison via Partitioning the Dynamic Range," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., Year: 2015, Volume: 24, Issue : 5.

[6] P. Patronick and S. J. Piestrak, "Design of Reverse Converters for the New RNS Moduli set {2n+1,2n−1,2n,2n−1+1} (n odd)," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, March. 2014.

[7] P. Patronick and S. J. Piestrak, "Design of Reverse Converters for the New RNS Moduli set {2n+1,2n−1,2n,2n−1+1} (n odd)," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, March. 2014.

[8] Y. Liu and E. M.-K. Lai, "Design and implementation of an RNS-based 2-D DWT processor," IEEE Trans. Consum. Electron., vol. 50, no. 1, pp. 376–385, Feb. 2004.

[9] N. S. Szabó and R. Tanaka, Residue Arithmetic and Its Applications to Computer Technology. New York, NY, USA: McGraw-Hill, 1967.

[10] M. A. Soderstrand, W. K. Jenkins, G. A. Jullien, and F. J. Taylor, Eds., Residue Number System Arithmetic: Modern Applications in Digital Signal Processing. New York, NY, USA: IEEE Press, 1986.

[11] C. B. Dutta, P. Garai, and A. Sinha, "Design of a reconfigurable DSP processor with bit efficient residue number system," Int. J. VLSI Design Commun. Syst., vol. 3, no. 5, pp. 175–189, Oct. 2012.

[12] M. Esmaeildoust, D. Schinianakis, H. Javashi, T. Stouraitis, and K. Navi, "Efficient RNS implementation of elliptic curve point multiplication over GF(p)," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 8, pp. 1545–1549, Aug. 2013.

[13] J. C. Bajard, N. Meloni, and T. Plantard, "Efficient RNS bases for cryptography," in Proc. 17th IMACS World Congr., Sci. Comput., Appl. Math. Simulation, Paris, France, Jul. 2005, pp. 1–7.

[14] J. C. Bajard, N. Meloni, and T. Plantard, "Efficient RNS bases for cryptography," in Proc. 17th IMACS World Congr., Sci. Comput., Appl. Math. Simulation, Paris, France, Jul. 2005, pp. 1–7.

[15] J. C. Bajard, N. Meloni, and T. Plantard, "Efficient RNS bases for cryptography," in Proc. 17th IMACS World Congr., Sci. Comput., Appl. Math. Simulation, Paris, France, Jul. 2005, pp. 1–7.

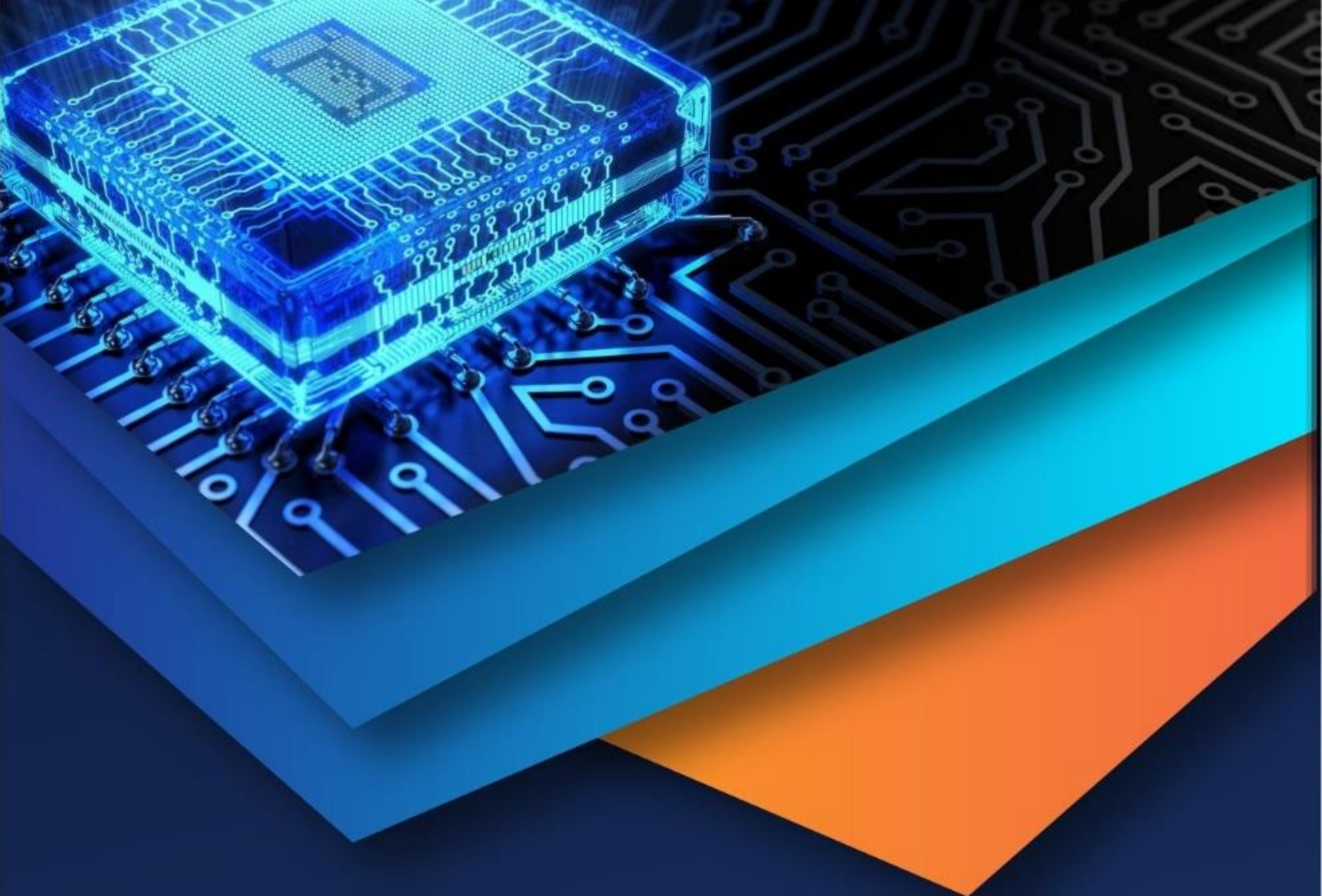
[16] A.SkavantzosandM.Abdallah,"Novelresiduearithmeticprocesso

[17] forhighspeeddigitalsignalprocessing,"inProc.32nd AsilomarConf. Signals, Systems,Computers, 1998, vol. 1, pp. 187–193.

[18] P.Fernandez,A.Garcia,J.Ramirez,L.Parrilla,andA.Lloris,"ARNSbasedmatrix-vector multiplyFCTarchitectureforDCTcomputation," in Proc. 43rd IEEE Midwest Symp. Circuits Syst., 2000, vol. 1, pp. 350–35

[19] J.Ramirez,A.Garcia,U.Meyer-Baese,andA.Lloris,"FastRNS-FPL based communications receiver design and implementation," in Proc. Int.Conf.FieldProgrammableLogic,2002,vol.2438,pp.472–481.

[20] J.-C. Bajard and L. Imbert, "A full RNS implementation of RSA," IEEETrans.Computers,vol.53,no.6,pp.769 774,Jun.2004.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY