iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 Issue: V Month of publication: May 2023

DOI: https://doi.org/10.22214/ijraset.2023.53019

Residue to Binary Converter for a Four Moduli Set

Priyank Tomar[1], Aditi Kashyap[2], Rishabh Anand[3], Ritesh Kumar Jaiswal[4]
[1, 2, 3]B.Tech Student, [4]Assistant Professor, Department of Electronics and Communication Engineering, JSS Academy of Technical Education, Noida ,201301, India

Abstract: Due to issues with conventional number systems (binary), such as the carry propagation issue, which reduces the performance of hardware computing systems, residue number systems (RNS) have gained popularity. Residue to binary conversion speed and complexity, as well as the speed of the internal RNS arithmetic unit, all affect how well a residue number system (RNS) performs. We have included a number of studies on the recognition of RNS Number Systems and Residue to Binary Converter Systems for various moduli sets in this paper. Additionally, we implemented a model and calculated the value of LUT'S, Slices and other components within it.

Keywords: RNS (Residue Number System), CRT (Chinese Remainder Theorem), MRC (Mixed Radix Conversion), Computer Arithmetic, CPA (Carry Propagation, Adder), CSA (Carry Save Adder).

I. INTRODUCTION

A unique and unweighted arithmetic number representation is the residue number system (RNS). A collection of co-prime integers is used to represent numbers that fall inside a specific dynamic range. It uses a carry-free mechanism for its mathematical representation, RNS. This suggests that there won't be any carry that spreads among the bits of the same digit. This resulted in an arithmetic form that is computationally quick since carry need not be propagated across multiple digits.[1]

This crucial RNS characteristic is only applicable to addition, subtraction, and multiplication. Other crucial function, such as the conversion of residue to base two system, sign analogy, scaling, etc. require more time than equivalent operations in base two arithmetic.[2] Although RNS is not majorly used in general-purpose processors, application-specific processors that primarily rely on basic operations like add, subtract and multiply can benefit from RNS's carry-free characteristic. [3][4]

As a result, RNS has been effectively employed in a number of high-speed computation-required applications, including communication systems, digital signal and image processing, digital sweep oscillators, digital filters, embedded systems, parallel DNA calculations, and cryptographic systems. RNS has gained appeal in new applications including deep neural networks, memory processing, efficient video coding, and current cryptosystems in order to enhance the value of DSPs.[5][6]

The crucial requirements that make RNS a feasible choice to integrate demanding applications in embedded systems and Internet of Things Devices are to improve performance, reduce power consumption, and increase dependability.[7]

To have more successful RNS-based execution of such applications, they must have sufficient parallelism and dynamic range. Class-1 and Class-2 are the two basic subcategories of parallelism. One modulus in Class-1 must be a power of 2, while the multiplication of the other two moduli should be a Mersenne number. With an extra modulus of m4, the class-2 is a kind of a parallel extension of a kind of Class-1.[8][9] It is not necessary that multiplying all the moduli of class-2 RNS will give a Mersenne number. Despite its many benefits, the reverse converter's complicated design results from the inter-modular calculations it uses. The primary constituents of RNS based processors are the base two to RNS converter, modulo arithmetic units, and reverse converters. The reverse converter has a complicated, non-modular construction compared to other components.[10] Its design needs to be given more consideration in order to avoid sluggish performance and jeopardizing the advantages of the RNS number system.

The performance of the planned reverse converter is affected by both the properties of the selected moduli set and conversion algorithm. As a result, many moduli sets have been created, and the performance of the RNS depends heavily on the hardware choice.

II. SELECTION OF MODULI SET

The choice of the moduli set in the Residue Number System (RNS) is an important factor that affects the efficiency and accuracy of arithmetic operations. Here are some properties of the moduli set in RNS:

1) *Coprimality:* The moduli in the RNS moduli set should be pairwise coprime (i.e., they should have no common factors other than 1). This property ensures that every residue combination corresponds to a unique integer, which is essential for accurate arithmetic operations.

2) *Large Moduli:* larger moduli in the moduli set allow for larger representable integers, which is desirable in many applications. However, larger moduli also require more complex arithmetic operations and larger storage requirements.
3) *Prime Moduli:* Prime moduli are often preferred in RNS because they offer better properties for arithmetic operations. For example, addition and subtraction operations in RNS with prime moduli can be performed using simple bitwise operations.
4) *Balanced Moduli:* A balanced moduli set consists of moduli that are close in value to each other. This property can help in significant reduction of the residues dynamic range. This will also help to streamline the arithmetic operations.
5) *Reduced Moduli Set:* A reduced moduli set consists of a subset of the original moduli set that can represent the same range of integers. A reduced moduli set can improve the efficiency of arithmetic operations by reducing the number of moduli required and the complexity of modular arithmetic.

Overall, a trade-off between accuracy, efficiency, and complexity must be taken into account when selecting the moduli set values in RNS which depends over the application's individual needs.

III. METHODS for RNS to BASE-2 CONVERSIONS

If we have to metamorphose a number from Residue Number System (RNS) to base two (Binary) system, the following steps need to be performed: -

Determine the RNS moduli set: In RNS, the number is represented in terms of residues modulo a set of coprime integers. For example, if we have moduli set {2, 3, 5}, then the RNS depiction of a number X would be (X mod 2, X mod 3, X mod 5).

Compute the product of all moduli in the moduli set. This product is denoted by M.

Compute the inverse of each modulus with respect to M using the Extended Euclidean Algorithm. The inverse of modulus mi with respect to M is denoted by Mi.

Multiply each residue in the RNS representation by its corresponding Mi value.

Add all the products obtained in step 4.

The sum obtained in step 5 is the binary equivalent of the original number.

Let's take an example to illustrate the conversion process from RNS to Binary system:

Suppose we have a number in RNS with moduli set {2, 3, 5}. The RNS depiction of the number is (1, 2, 3).

Compute the product of all moduli: M = 2 x 3 x 5 = 30

Compute the inverse of each modulus with respect to M using the Extended Euclidean Algorithm:

Inverse of 2 with respect to 30: 2 x 15 = 1 (mod 30), so Mi = 15

Inverse of 3 with respect to 30: 3 x 10 = 1 (mod 30), so Mi = 10

Inverse of 5 with respect to 30: 5 x 6 = 1 (mod 30), so Mi = 6

Multiply each residue by its corresponding Mi value:

1 x 15 = 15

2 x 10 = 20

3 x 6 = 18

Add all the products obtained in step 3: 15 + 20 + 18 = 53

The sum obtained in step 4 is the binary equivalent of the original number. Converting 53 to binary, we get 110101.

Therefore, the RNS representation (1, 2, 3) is equivalent to the binary number 110101.

IV. LITERATURE SURVEY

[10] An innovative approach for the implementation of modulo based operations which are required for CRT was provided in this work. We may identify the multiple of the modulus that must be deducted by including a product with 2k of the items in the summation and afterwards only examine the k number of bits which are least significant for us. The technique is also used with adder tree-based CRT systems, within which the aggregate results are reckoned with the help of a variety of dispersed structures. Regarding speed and space occupancy, the resulting architecture seems to be highly appealing. A 32-bit CRT architecture is currently being developed for extremely fast DSP applications. ES2 CMOS 0.7 p technology is being used to create the converter.

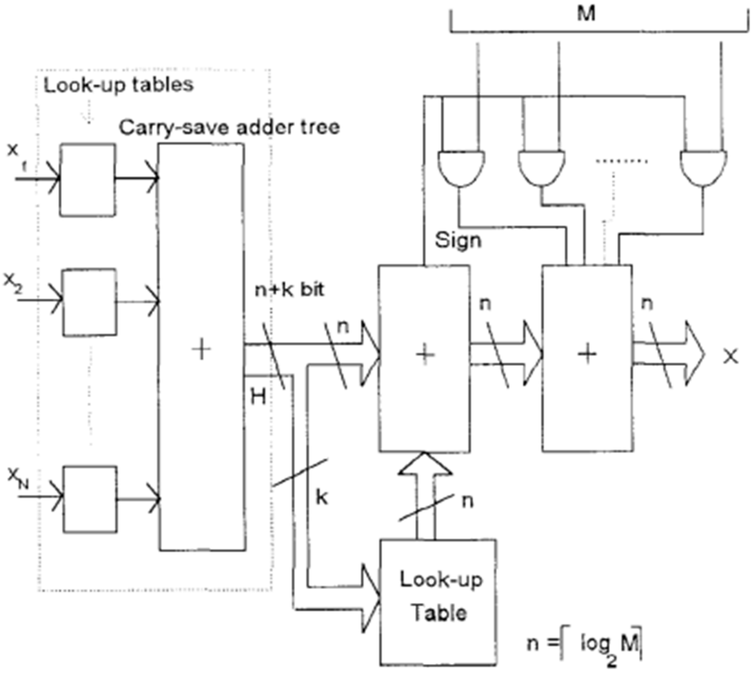


Fig-1: Block Diagram for the Design mentioned in [10]

[11] This work presents three possible residue-to-base two converters of the moduli set (2n-1, 2n, 2n+1). The novel converters suggested here perform better in terms of speed and area when compared to the earlier converters. The newly developed New Chinese Remainder Theorems are the foundation upon which the new converters are built. The New Chinese Remainder Theorems are anticipated to have a substantial impact on future residue-to-base two converter design for various moduli sets.

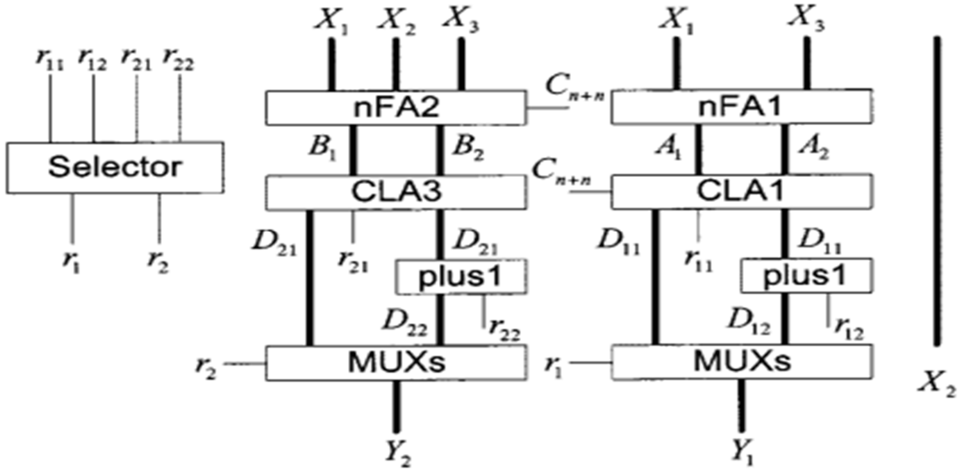


Fig-2: Block Diagram for the design mentioned in [11].

[12] An enhanced breakneck implementation of a residue-to-base two converter for RNS (2^n + 1, 2^n, 2^n- 1) was suggested in this study. Adders made up the complete structure. But because only two stages of complete adders and only a single 2n bit 1's complement adder account for its whole latency, it is substantially quicker. The entire stretch of the operands presented to the single-stage adder which is working in the carry-propagate mode within this residue-to-base two converter is just two-third in value as compared to the full range of RNS.As a result, the RNS gains an extra significant benefit from a fantastic residue-to-binary conversion time made possible by effective hardware.

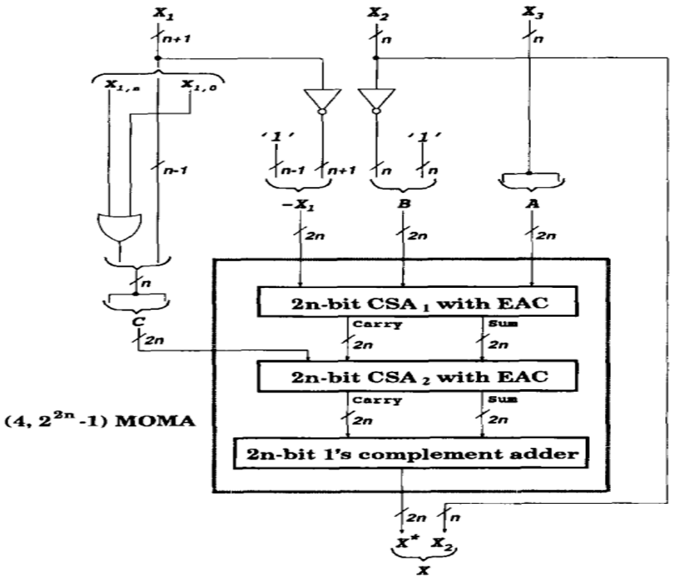


Fig-3: Block Diagram for the design mentioned in [12].

[13] A residue to base two converters for the moduli set (2n-1, 2n, and 2n+1) was reported in this work, and it reduced the latency for the best reverse converter for the same moduli set by a factor of 2. A tweak that would lower the number of full adders employed in the architecture by a factor of (n-1) was also suggested in order to accomplish this considerable performance advantage with almost no increase in hardware requirements. For the provided moduli set, the reverse converter that is offered is the quickest and requires the least amount of hardware. Additionally, it addresses the issue of the duplicate representation of zero without adding any additional hardware.

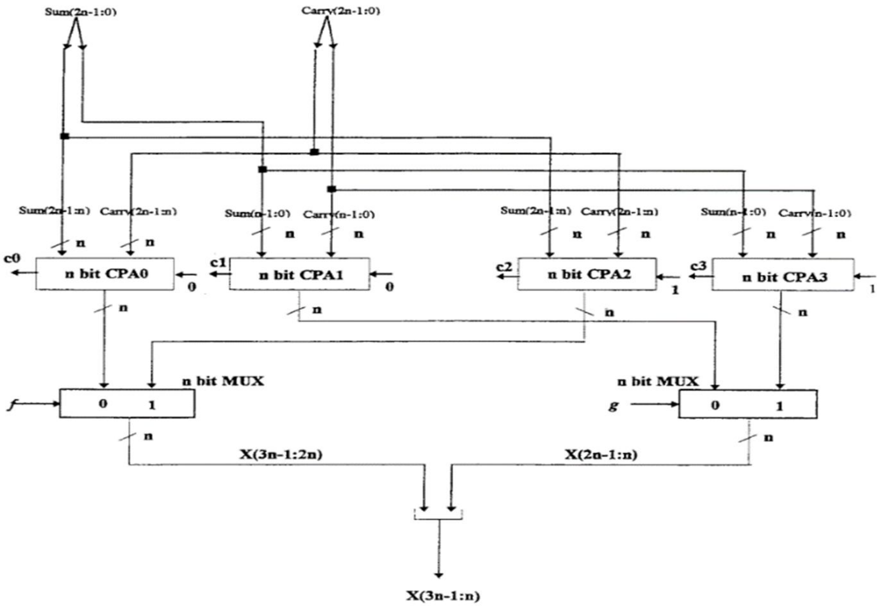


Fig-4: Block Diagram for the design mentioned in [13].

[16] In this study, a three-module RNS was analysed using an RNS balancing metric, the CRT constants for RNS were computed, a device to accomplish the conversion for RNS was proposed, and the effectiveness of the proposed reverse conversion was verified. According to the simulation findings, the suggested conversion process required less amount of hardware (10-15%) and is faster (2-25%). Certain various techniques can be used to make the conversion. Reverse conversion is possible with minimal hardware costs and great speed because to the usage of inexpensive modules.

Table- 1: Summary of Literature

Paper	Author	Techniques	Advantages
[1]	Ali Akbari, Mehdi Hosseinzadeh, Shiva TaghipourElvazi.	Dynamic range division and the Chinese Remainder Theorem.	In comparison to competing converters, the converter obtained an average area time saving of 20%.
[2]	Ritesh Kumar Jaiswal, Raj Kumar and Ram Awadh Mishra.	Mixed Radix Conversion, and Carry Save Adders.	For a 5n-bit dynamic range, more speed and less hardware complexity are required.
[3]	Amani Goniemat, Andraws Swidan.	Implemented using the CRT-1 technique.	For n=4, it had a dynamic range it had a DR of 22 bits and a delay of 10n+6.
[4]	M. R. Noorimehr, M. Hosseinzadeh, R. Farshidi	Implemented using the new Chinese remainder theorem.	Compared to comparable reverse converters with 4n-bit DR, this converter has a shorter latency and a cheaper hardware cost.
[5]	Mohammad Reza Taheri, Nasim Shafiee, Mohammad Esmaeildoust, Zhale Amirjamshidi, Reza Sabbaghi-nadooshan , Keivan Navi	MRC (Mixed Radix Conversion)	The efficient Arithmetic Unit (AU) is a feature of this set of moduli. reduced unit delay and gate area.
[6]	Ahmad Hiasat	CRT Technique	Significantly reduced area by 9.9%, standby time by 16.9%, and power consumption by 10.6%.
[7]	Sheba Diamond Thapa, Mridupawan Sonowal, Prabir Saha.	CRT, MRC and New CRT-1.	
[8]	Azadeh Alsadat Emrani Zarandi, Samuel Antao, Leonel Sousa.	CRT and parallel-prefix excess one (HMPE) adder.	The VLSI implementation show significant delay reduction and area x time ^2 improvements.
[9]	K. A Gbolagade, R. Chaves, S.D Cotofana, L.Sousa.	CRT with Carry Save Adders and two Carry Propagate Adders.	The proposed converter achieves about 19% delay reduction.
[10]	Ahmad Hiasat	CRT and MRC	The proposed structure improves area, time, and power characteristics by 22.5%, 24.6%, and 20.5%, respectively.

V. RNS to BINARY CONVERSION TECHNIQUES

Several residues to binary conversion techniques can be used for the conversion of residue number system to base two number system depending upon the moduli set, complexity, speed, power consumption etc. required. Some of these techniques are being used are CRT (Chinese Remainder Theorem), MRC (Mixed Radix Conversion).

A. Chinese Remainder Theorem (CRT)

The base two number X which corresponds to the residues (x1, x2, x3, ..., xn) in the RNS representation {m1, m2, m3, ... mn} is calculated by CRT using: -

$$X=((x_1(1/M_1)_{m1})_{m1} *M1 + (x_2(1/M2)_{m2})_{m2}*M2+............ (x_n (1/M_n))_{mn} *M_n) *mod\ M.$$

The quantities here $(1/M_j)_{Mj}$ are known as the multiplicative inverse of m_j mod m_j which is described such that $M_j(1/M_j)m_j=1$.

The major advantage of the CRT is that the weighting and summation of all the residues x_i can effectively be done in parallel, followed by reduction mod M.

CRT is used efficiently in the case of three or four moduli sets since n bits of the decrypted number X can be used directly in the form of residues which are correspondent to modulus 2^n and the multiplication of all the remaining moduli which requires modulo reduction can be efficiently used in the case of the different three or four moduli sets.

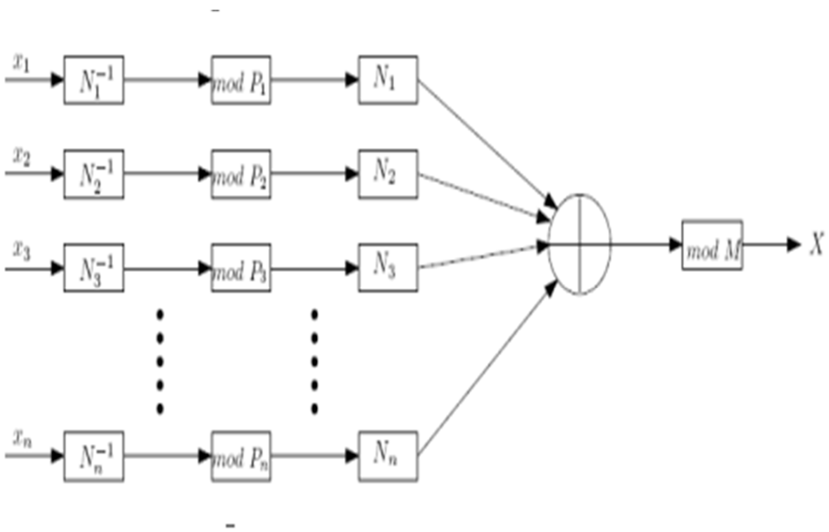


Fig-5: CRT Block Diagram implementation.

B. Mixed Radix Conversion (MRC)

The sequential MRC approach incorporates multiplicative inverses of a single modulus with regard to the other moduli, as well as modulo subtractions and modulo multiplications. The decrypted number is written as follows in MRC: -

$A = x + d1m1 +d2m2 +....... D_{j-1}m1m2...m_{j-1.}$

Mixed Radix Digits are the name given to the parameters di. One mixed radix digits di is established in each phase. The MRC digits are weighted at the end, and this weighting is utilised to get the final decoded number. The last modulo reduction is not necessary to complete.

To ensure that the outcome is exactly divisible by each modulus, it should be noted that the residue associated with each step is removed. This division is accomplished via multiplication with multiplicative inverse.

In the last stage, larger integers must be multiplied, such as z1m2m3 in the example of the three moduli, and these products must then be added using carry-save-adders before being added using CPA. However, since the outcome which is obtained from MRC is always smaller than M=m1m2m3, there is no need for modulo reduction in the end.

Pipelined architecture is also possible while working with MRC.

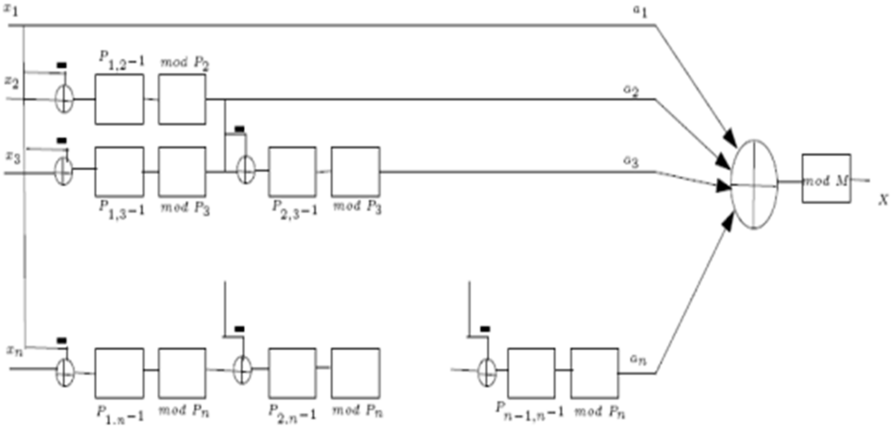


Fig-6: MRC Block Diagram implementation.

C. NEW CRT-1

The pre-existing CRT method has undergone several modifications in an effort to provide better results, with the most crucial of them being the New CRT-1 approach for the Residue to Binary Conversion. The weighted base two number which corresponds to the residues (x_1, x_2, x_3..., x_n) may be determined with the help New CRT-I by providing the moduli set {m_1, m_2, m_3..., $m_{n\}}$ through: -
X= $x_1 + (k_1\ m_1\ (x_2-x_1) + k_2\ m_2(x_3-x_2) +$…...$k_{(n-1)}\ m_1\ m_2\ m_3 \dots m_n\ (x_n-x_{n-1}))$ mod $(m_2\ m_3 \dots m_{n-1}\ m_n)$.

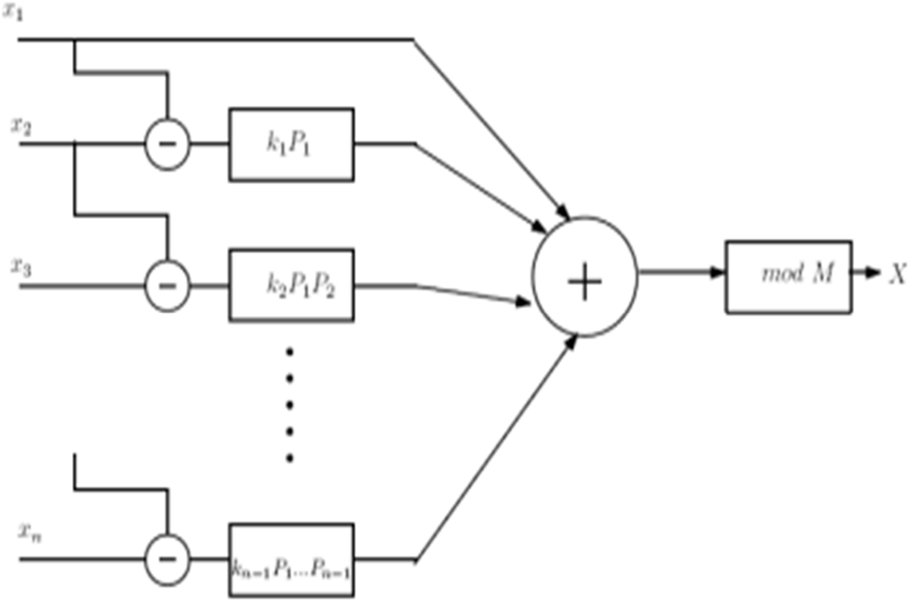


Fig-7: NEW CRT-1 Block Diagram implementation.

D. NEW CRT- 2

The New Chinese Remainder Theorem 2 states that the needed number X may be determined from its residues {x_1, x_2, and x_3} using the following process: -

$$A = B + P_1P_2 \left|k_1\ (C-B)\right|_{p3p4}$$
$$B = x_1 + P_1 \left|\ k_2(x_2-x_1)\right|_{p2}$$
$$C = x_3 + P_3 \left|k_3(x_4-x_3)\right|_{p4}$$

k_1, k_2 and k_3 are the value of the multiplicative inverse.

VI. IMPLEMENTED MODEL RESULTS

If we want to design an efficient RNS based processor unit then it is very much required to design an effective and accurate residue to base 2 converters. Here we are going to design an efficient residue to base 2 converters for our moduli set $\{2^n - 1, 2^n + 1, 2^{2n} + 1, 2^{2n+p}\}$.

Some of the conventions which are used for the development of this design are: -

1) $\{z_1, z_2, z_3, z_4 \ldots\ldots\ldots z_N\}$ represents moduli set of N length which are relatively prime to each other.
2) $M = \prod_{i=1}^{N} z_{i,\text{ it}}$ represents the dynamic range of a reverse converter.
3) For any number A, if it belongs to the dynamic range of the converter, then the residue representation if the number can be given as $A \rightarrow (S_1, S_2, S_3, \ldots\ldots\ldots S_N)$ where P_i is the value of the smallest positive remainder which is obtained when A is divided by m_i.
4) $\hat{z}_i = (M/ z_i)$.

Here, we have used the Chinese Remainder Theorem for the development of the converter structure for the four moduli set $\{z_1, z_2, z_3, z_4\}$ where $z_1 = 2^n - 1$, $z_2 = 2^n + 1$, $z_3 = 2^{2n} + 1$ and $z_4 = 2^{2n+p}$. The calculation through CRT can be proceeded as: -

$$A = \sum_{i=1}^{4} \hat{z}_i \langle 1|\hat{z}_i\rangle_{z_i} P_i$$

The residue representation of A can be written as (S_1, S_2, S_3, S_4) where the value of A belongs to the dynamic range of the converter. These four digits i.e. (S_1, S_2, S_3, S_4) can be represented into their equivalent binary representation as: -

1) $S_1 = \sum_{i=0}^{n-1} s_{1(i)} 2^i = s_{1(n-1)} \ldots\ldots\ldots s_{1(0)}$
2) $S_2 = \sum_{i=0}^{n} s_{2(i)} 2^i = s_{2(n)} \ldots\ldots\ldots s_{2(0)}$
3) $S_3 = \sum_{i=0}^{2n} s_{3(i)} 2^i = s_{3(2n)} \ldots\ldots\ldots s_{3(0)}$
4) $S_4 = \sum_{i=0}^{2n+p-1} s_{4(i)} 2^i = s_{4(2n+p-1)} \ldots\ldots\ldots s_{4(0)}$.
5) Now we have to calculate the value of $\hat{z}_i$ in order to find the multiplicative inverse to use within the Chinese Remainder Theorem. The value of $\hat{z}_i$ can be calculated as $\hat{z}_i = (M/ z_i)$ where $\{i= 1,2,3,4\}$.
6) Now when we calculate the value of $\hat{z}_i$ for the design it is obtained as: - $\hat{z}_1=(2^n+1)(2^{2n}+1)2^{2n+p}$, $\hat{z}_2 = =(2^n-1)(2^{2n}+1)2^{2n+p}$, $\hat{z}_3 = (2^{2n}-1)2^{2n+p}$ and the value of $\hat{z}_4 = (2^{4n}-1)$.
7) The multiplicative inverses now can be calculated for the moduli set values. The value of multiplicative inverse is given as: - $\langle(1/ \hat{z}_1\rangle_{z_1}= 2^{n-p-2}$, $\langle(1/ \hat{z}_2\rangle_{z_2}= 2^{n-p-2}$, $\langle(1/ \hat{z}_3\rangle_{z_3}=(-2^{2n-p-1})$, $\langle(1/ \hat{z}_4\rangle_{z_4}=(-1)$.
8) These values of multiplicative inverse obtained in the above step can be used to utilize the CRT for the conversion process.
9) When we put these values within the CRT equation the value for A is given by: -

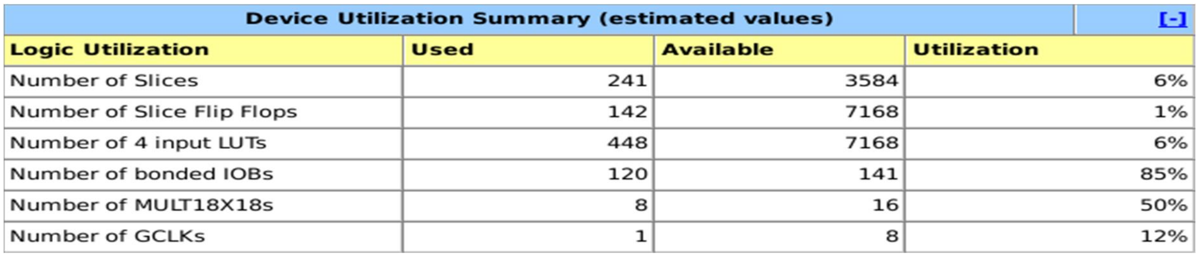
$$A = (2^n + 1)(2^{2n} + 1)2^{2n+p}2^{n-p-2}P_1 + (2^n - 1)(2^{2n} + 1)2^{2n+p}2^{n-p-2}P_2 - (2^n - 1)2^{2n+p}2^{2n-p-1}P_3 - (2^{4n} - 1)P_4$$

Now by dividing the above equation with 2^{2n+p}, taking the floor value and taking 2^{n-p-2} as common from the entire equation, and applying $(2^{4n} - 1)$ the above equation can be represented as: -

$$\lfloor A/ 2^{2n+p} \rfloor = \langle 2^{n-p-2}(t_1 + t_2 + t_3 + t_4\rangle_{(2^{4n}-1)}$$

For the simulation of our designed converter, we have used the ModelSim Altera 6.4a (Quartus II 9.0) edition. For the implementation of the converter total six number of modules have been created including the testbench module for the converter design.

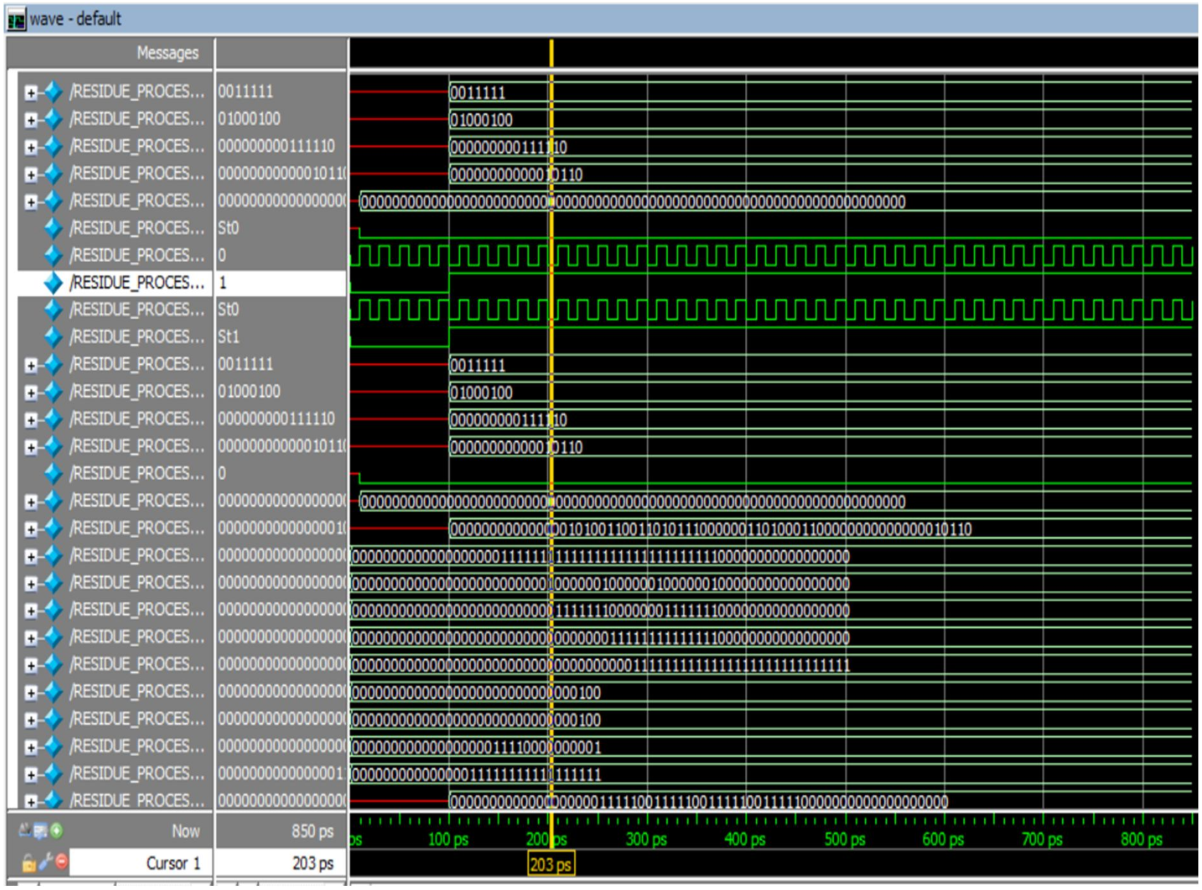
The Fig. 8 shows the synthesis result of the designed converter. It specifies a various number of factors for the converter such as the number of slices used in the converter, number of LUT's used for the implementation of the converter, number of slice flip flops etc. The following results are obtained with the help of Xilinx synthesis tool. The total value of gate delay for the converter was obtained to be 42.891 ns.



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	241	3584	6%
Number of Slice Flip Flops	142	7168	1%
Number of 4 input LUTs	448	7168	6%
Number of bonded IOBs	120	141	85%
Number of MULT18X18s	8	16	50%
Number of GCLKs	1	8	12%

Fig-8: Synthesis of the converter

The converter has been implemented for a value of n=7 within the given moduli set. The Fig. 9 shows the simulation results for the designed residue to base 2 converters.



wave - default

Messages	
/RESIDUE_PROCES...	0011111
/RESIDUE_PROCES...	01000100
/RESIDUE_PROCES...	000000000111110
/RESIDUE_PROCES...	00000000000010110
/RESIDUE_PROCES...	0000000000000000
/RESIDUE_PROCES...	St0
/RESIDUE_PROCES...	0
/RESIDUE_PROCES...	1
/RESIDUE_PROCES...	St0
/RESIDUE_PROCES...	St1
/RESIDUE_PROCES...	0011111
/RESIDUE_PROCES...	01000100
/RESIDUE_PROCES...	000000000111110
/RESIDUE_PROCES...	00000000000010110
/RESIDUE_PROCES...	0
/RESIDUE_PROCES...	0000000000000000
/RESIDUE_PROCES...	00000000000000010
/RESIDUE_PROCES...	0000000000000000
/RESIDUE_PROCES...	0000000000000000
/RESIDUE_PROCES...	0000000000000000
/RESIDUE_PROCES...	0000000000000000
/RESIDUE_PROCES...	0000000000000000
/RESIDUE_PROCES...	0000000000000000
/RESIDUE_PROCES...	0000000000000000
/RESIDUE_PROCES...	0000000000000000
/RESIDUE_PROCES...	00000000000000011
/RESIDUE_PROCES...	0000000000000000
Now	850 ps
Cursor 1	203 ps

Fig-9: Simulation of Converter

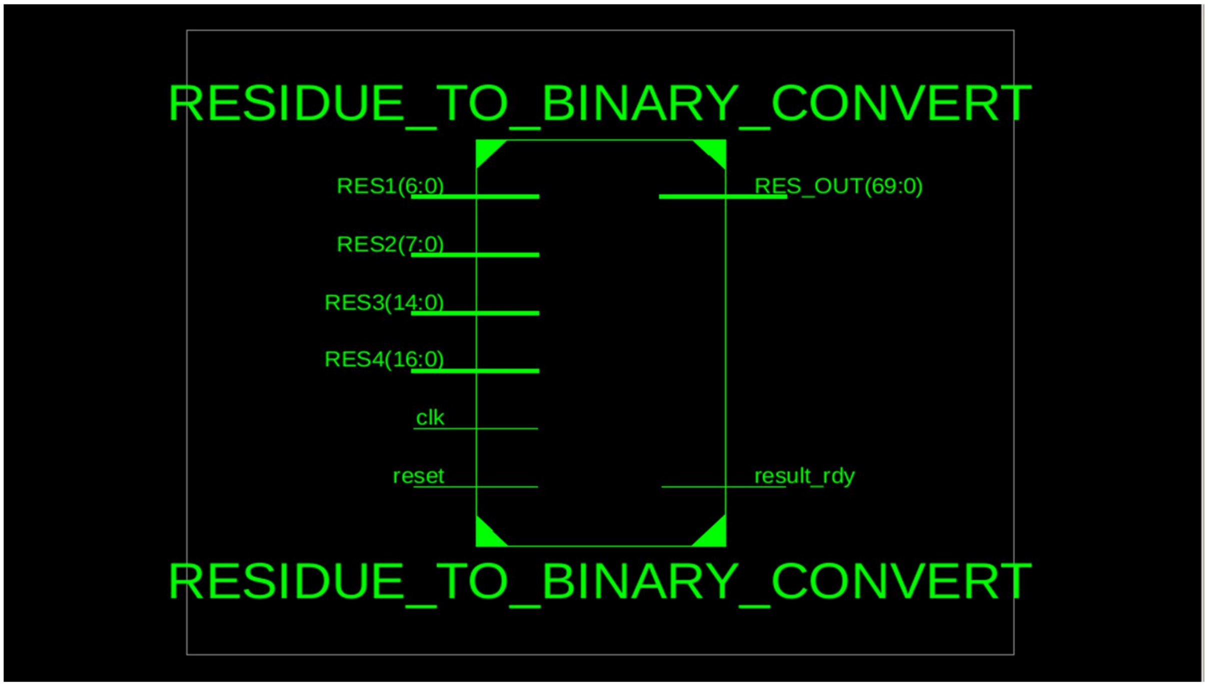


Figure 4.10: RTL of the converter.

VII. CONCLUSION

The Residue Number System (RNS)'s intrinsic parallelism and modular arithmetic capabilities are extremely advantageous for the development of application-specific digital signal processors (DSP). Along with providing value to DSPs, RNS is becoming more and more popular in the construction of cutting-edge applications like deep neural networks, high efficiency video coding, contemporary cryptosystems, and memory processing [6]. RNS has several applications in various fields, and its advantages in terms of speed and efficiency make it a useful tool in many applications. RNS is a strong contender for integrating a variety of applications which requires higher amount of computation into embedded systems and IOTdevices by enhancing performance, reducing power consumption, and also improving dependability. This paper outlines a variety of methods for successfully converting the residue number system to base two number systems.

REFERENCES

[1] Ali Akbari, Mehdi Hosseinzadeh, Shiva TaghipourEivazi, Somayyeh Jafarali Jassbi a New High-Speed, Low-Area Residue-to-Binary Converter for the Moduli Set $\{2^{4n}, 2^{2n} + 1, 2^n + 1, 2^n - 1\}$ Based on CRT-1 Circuits, Systems, and Signal Processing 40:5773–5786, (2021).

[2] R.K. Jaiswal, R. Kumar, R.A. Mishra, Area efficient memoryless reverse converter for new four moduli set $\{2^{n-1}, 2^n - 1, 2^n + 1, 2^{2n+1}-1\}$. J. Circuits Syst. Comput. 27, 1–13 (2018)

[3] Amani Goniemat, Andraws Swidan, A High-Speed Residue-to-Binary Converter for Three-Moduli Set, IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 4, No 1, July. (2013).

[4] M.R. Noorimehr, M. Hosseinzadeh, R. Farshidi, High speed residue to binary converter for the new four-moduli set $\{2^{2n}, 2^n + 1, 2^{n/2} + 1, 2^{n/2}-1\}$. Arab. J. Sci. Eng. 39, 2887–2893 (2014)

[5] Mohammad Reza Taheri, Nasim Shafiee, Mohammad Esmaeildoust, Zhale Amirjamshidi, Reza Sabbaghi-nadooshan, Keivan Navi, A High-Speed Residue-to-Binary Converter for Balanced 4-Moduli Set. Journal of Computing and Security Volume 2, Number 1 (pp. 43–54), January (2015).

[6] A. Hiasat, A residue-to-binary converter with an adjustable structure for an extended RNS three moduli set. J. Circuits Syst. Comput. 28, 1950126 (2018).

[7] Sheba Diamond Tabaha, Mridupawan Sonowalb , Prabir Sahaa,b, On the design of efficient Redidue to Binary Converters, International Conference on Computational Intelligence and Data Science (ICCIDS).(2018).

[8] Azadeh Alsadat Emrani Zarandi, Amir Sabbagh Molahosseini, Mehdi Hosseinzadeh, Saeid Sorouri, Samuel Antão, and Leonel Sousa Reverse Converter Design via Parallel-Prefix Adders: Novel Components, Methodology, and Implementations, IEEE TRANSACTIONS ON VERY LARGE-SCALE INTEGRATION (VLSI) SYSTEMS. (2015)

[9] K.A. Gbolagade, R. Chaves, L. Sousa, S.D. Cotofana,An Improved RNS Reverse Converter for the Moduli Set , 978-1-4244-5309-2/10/$26.00 ©2010 IEEE.(2010)

[10] **G.C. Cardarilli, M. Re, R. Lojacono, EFFICIENT MODULO EXTRACTION FOR CRT BASED RESIDUE TO BINARY CONVERTER, 1997 IEEE International Symposium on Circuits and Systems, - June 9-12,1997, Wong Kong.

[11] S. Andraros and H. Ahmad, “A new efficient memoryless residue to binary converter,” IEEE Trans. Circuits Syst., vol. CAS-35, pp. 1441–1444, Nov. 1988.

[12] Stanislaw J. Piestrak, A High-speed Realization of a Residue to Binary Number System Converter, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-11: ANALOG AND DIGITAL SIGNAL PROCESSING, VOL. 42, NO. 10, OCTOBER 1995.

[13] M. Bhardwaj, A. B. Premkumar, and T. Srikanthan, Breaking the -Bit Carry PropagationBarrier in Residue to Binary Conversion for the Modula Set, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: FUNDAMENTAL THEORY AND APPLICATIONS, VOL. 45, NO. 9, SEPTEMBER 1998.

[14] . A. Bayoumi, G. A. Jullien, and W. C. Miller, “A VLSI implementation of residue adders,” IEEE Trans. Circuits Syst., vol. 34, pp. 284–288, 1987.

[15] K. M. Elleithy and M. A. Bayoumi, “A (1) algorithm for modulo addition,” IEEE Trans. Circuits Syst., vol. 37, pp. 628–631, May 1990.

[16] A. Hiasat, A residue-To-binary converter for the extended four-moduli set $\{2^n - 1, 2n + 1, 2^{2n+1}, 2^{2n+p}\}$. IEEE Trans. Very Large-Scale Integration. Syst. 25, 2188–2192 (2017)