International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 161-173
Published Online March 2024 in IJEAST (http://www.ijeast.com)

OPTIMUM THREE MODULI SET FOR ANY WORD BIT LENGTH N AND I = 0, 1, 2

Gurinder Bawa
Stellantis, Auburn Hills, MI, USA

Abstract—**In Residue Number System (RNS) the most important factor is the choice of optimum moduli set. The most used three moduli set is $M_0 = (2^n -1, 2^n, 2^n +1)$, but it is only optimum when the word bit length N is exactly divisible by 3. A new three moduli set is proposed $(2^n -1, 2^{n+i}, 2^n +1)$, where $i = |N|_3$. It is shown to have the same dynamic range efficiency as M_0 for any N. This new Moduli set is implemented using MRC II and CRT. The CRT implementation is shown to have same architecture for any i. The hardware consists of 6n FA, 2n + 1 + i inverters with propagation delay of 2n + 2.**

Keywords—**Residue to Binary Converters, Residue Number System, Chinese Remainder Theorem II.**

I. INTRODUCTION

RNS has attracted many researchers for the past few decades that involve extensive computation in the field of signal processing such as fast Fourier Transform (FFT), Image Processing, and Digital Filters [1],[2]. High speed and less complexity of hardware requires the large number to be split into smaller numbers so that instead of one large computation several small computations can be executed in parallel which reduces the load and time of calculation. Thus, choice of proper moduli set could greatly make the design efficient and cost effective. RNS moduli sets which are discovered till now are mostly in the form of power of 2's. Further the moduli set could be a group of 3, 4 or 5 depending upon the complexity and the dynamic range of system.

In this paper we proposed a new three moduli set which overcomes the limitation of conventional and mostly commonly used three moduli set $(2^n -1, 2^n, 2^n +1)$ [3][4]. In our work we considered three cases:

Let N be the Word Bit Length.

1. If $|N|_3 = 0$ then use moduli set $M_0 = (2^n -1, 2^n, 2^n +1)$ which is same as the most popular moduli set.
2. If $|N|_3 = 1$ then usemoduli set $M_1 = (2^n -1, 2^{n+1}, 2^n +1)$.
3. If $|N|_3 = 2$ then use moduli set$M_2 = (2^n -1, 2^{n+2}, 2^n +1)$.

Once the correct moduli set is chosen, the next target it to convert RNS into binary system so that RNS based computations could be interfaced with the real-world binary computers. In our work we implemented the ideas of M.Akkal and P. Siy [5] using MRC-II and Ahmad Hiasat and Andraos Sweidan [6] using CRT. In the later sections we have also concluded that our new moduli set is better in terms of efficiency.

II. BACKGROUND

There are different three moduli set that are proposed in the literatures, but the most popular three moduli set is $(2^n -1, 2^n, 2^n +1)$ [3]. Some other three moduli sets are (2n-l, 2n, 2n+l)[7], (2n, 2n+1, 2n+2)[8], $(2^n -1, 2^n +1, 2^{2n} +1)$[11],$(2^{k-1}-1, 2^k -1, 2^k)$ [9],$(2^n – 1, 2^n + 1, 2^{2n+1})$ [10], $(2^n, 2^n – 1, 2^{2n-1} – 1)$ [12], $(2^{n-1}, 2^n -1, 2^n +1)$[6]have been proposed. Even all of the above-mentioned moduli sets have been proposed and claims some advantages, but no study yet has been carried out to compare these moduli sets.

A. **Characteristics of Moduli Set:** Dynamic Range and Dynamic Range Efficiency can be used to compare moduli set.

1) The **Dynamic Range** is defined as the product of moduli in a given set.

$$M = \prod_{i=1}^{3} m_i \qquad (1)$$

2) Dynamic range efficiency 'D_{eff}' is the ratio of dynamic range of the moduli set to dynamic range of word bit length.

$$D_{eff} = M / 2^N \qquad (2)$$

The following example will illustrate how those above-mentioned parameters are calculated. Let us consider word bit length N = 9 bit. According to M_0 the three moduli set is 2^n-1, 2^n, 2^n+1.

where n is calculated as $n = N / 3 = 9 / 3 = 3$ (3)

If we substitute the above calculated value of n as 3 then the three moduli sets are:

$m_1 = 2^3 - 1 = 7 m_2 = 2^3 = 8 \quad m_3 = 2^3 + 1 = 9$

a. Dynamic rangeaccording to (1) is given as:

$$= (2^n)* (2^n-1)*(2^n + 1) = 2^{3n} – 2^n = 504. \qquad (4)$$

b. Dynamic Range Efficiency (D_{eff}) according to (2) is:

$$= (2^{3n} – 2^n) / 2^{3n} = 1 – 1/ 2^{2n}$$

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 161-173
Published Online March 2024 in IJEAST (http://www.ijeast.com)

= 1 – 1/ 2^{2*3}= 1 – 1/ 64= .99 or 99%(5)

The efficiency of M_0 is the reason why M_0 is the most cited 3 moduli set. It will be shown that efficiency disappears if $|N|_3 \neq 0$.

1. Case1: $|N|_3 = 1$
ForN = 10, when divided by 3 gives us remainder 1 and
n =(N – 1) / 3
Value of n when calculated comes out to be 3 and the modulus set is 7, 8, 9.
Dynamic range according to (1) is **504**.
Dynamic range efficiency (D_{eff}), when calculated according to (2)

$$D_{eff} = {}^{504}/_{1024} \cong 50\% \quad (6)$$

This 3 Moduli set can only represents 50% of the desired dynamic range.

2. Case2: $|N|_3 = 2$
ForN = 11, which when divided by 3 gives remainder 2. Sothe value of n is calculated using and n = (N – 2) / 3
Calculating the value of n yields 3 and the modulus set is 7, 8, 9.
Dynamic range according to (1) comes out to be **504**.
Dynamic range efficiency (D_{eff}), when calculated according to (2):

$$D_{eff} = {}^{504}/_{2048} \cong 25\% \quad (7)$$

Again this 3 moduli set can only represents 25% of thedesired dynamic range.

III. CORRECTED MODULI SET

According to the most popular moduli set the three moduli set is $M_0 = 2^n$-1, 2^n, 2^n+1. But as we discussed in above section that it is no longer efficient if N is not exactly divisible by 3. In order to overcome these shortcomings, we had approached the problem in different way and thus come up with a unique approach of solving this problem. We took two cases again.

1. When bit word length 'N' divided by 3 gives remainder 1
2. When bit word length 'N' divided by 3 gives remainder 2.

The modified moduli set for:

1. **When** $|N|_3 = 1$

In this case we had increased the central modulus by one so that our new moduli set looks like $M_1 = 2^n$-1, 2^{n+1}, 2^n+1. We had further verified that this is the most efficient moduli set.

Example 3.1: Let us consider N = 10. The moduli set according to our modified formula is; $M_1 = 2^n$-1, 2^{n+1}, 2^n+1 is 7, 16, 9.

(a) Dynamic range: As defined in the section 2.1, the dynamic range is the product of all the modulus in the moduli set and is given as:

$$M = \prod_{i=1}^{3} m_i = (2^n - 1)* (2^{n+1})* (2^n + 1)$$
$$= (2^{n+1})*(2^{2n} - 1) = 2^{3n+1} - 2^{n+1} \quad (8)$$

N = 3n + 1:
n = (10 – 1) / 3 = 3

Substituting the value of n in (8)

$$M = (2^{3*3+1} - 2^{3+1}) = (1024 - 16) = 1008 \quad (9)$$

(b) Dynamic Range Efficiency D_{eff} is defined as the ratio of dynamic range (M) to dynamic range of bit word length.

$$D_{eff} = M / 2^N = \frac{(2^{3n+1} - 2^{n+1})}{2^{3n+1}}$$
$$= 1 - \frac{1}{2^{2n}} = 1 - \frac{1}{64} = .99 \text{ or } 99\% \quad (10)$$

This is same as equation (5).

1. **When** $|N|_3 = 2$

In this case we had increased the central modulus by two so that our new moduli set $M_2 = 2^n$-1, 2^{n+2}, 2^n+1. In order to verify that this is most efficient moduli set we took an example as shown below.
Example 3.2: Let us consider N = 11. The moduli set according to the modified formula is; $M_2 = 2^n$-1, 2^{n+2}, 2^n+1 is 7, 32, 9.

(a) Dynamic range: The dynamic range is the product of all the modulus in the moduli set and is given as:

$$M = \prod_{i=1}^{N} m_i = (2^{n+2})*(2^{2n} - 1) = 2^{3n+2} - 2^{n+2} \quad (11)$$

N = 3n + 2;
n= (11 – 2) / 3 = 3

And if we substitute the above value of n in (11) we get

$$M = (2^{3*3+2} - 2^{3+2}) = (2048 - 32) = 2016 \quad (12)$$

(b) Dynamic Range Efficiency D_{eff} is defined as the ratio of dynamic range (M) to dynamic range of bit word length.

$$D_{eff} = M / 2^N = \frac{(2^{3n+2} - 2^{n+2})}{2^{3n+2}} = 1 - \frac{1}{2^{2n}} = 1 - \frac{1}{64} = .99 \text{ or } 99\% \quad (13)$$

This is same as equation (5).

The results can be summarized in the following theorem.

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 161-173
Published Online March 2024 in IJEAST (http://www.ijeast.com)

Theorem 1: For anyword bit length N, the optimized 3 moduli set are given below:

1. If $|N|_3$= 0 so N = 3n, then Modlui set is given as M_0 = (2^n-1, 2^n, 2^n+1).
2. If $|N|_3$= 1 so N = 3n+1 then Modlui set is given as M_1 = (2^n-1, 2^{n+1}, 2^n+1).
3. If $|N|_3$= 2 so N = 3n+2 then Modlui set is given as M_2 = (2^n-1, 2^{n+2}, 2^n+1).

With $D_{eff} = 1 - \frac{1}{2^{2n}}$

IV. RNS TO BINARY CONVERTER

Once we have the correct moduli set, the next target undoubtedly should be to ascertain the appropriate RNS to Binary converter and then plugging the RNS arithmetic to Binary world. For that we have followed the work of M.Akkal and P. Siy [5] using MRC-II and Ahmad Hiasat and Andraos Sweidan [6] using CRT.

A. RNS to Binary Converter using MRC II

According to MRC II [5] number X is represented as:

$X_i = a_i M_{i-1} + X_{i-1}$ for $2 \le i \le n$; $n \ge 2$ (14)

Where a_i is the MRC coefficient, n is the number of moduli, $X_1 = a_1 = r_1$, $X_n = X$ and $M_{i-1} = \prod_{j=1}^{i-1} m_j$.Coefficient a_i can be computed using small look up tables. These tables are labeled as a_i. For n moduli set the number of look-up tables required are n – 1, with a size of $m_i \log_2 m_i$, where m_i is the i^{th} moduli. The functional block diagram for calculating X_i is shown in figure 1.

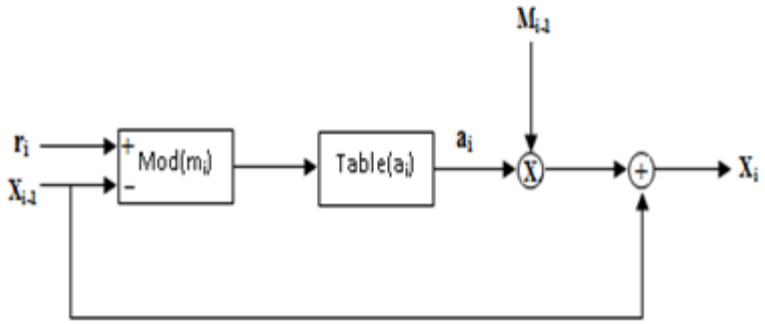


Fig. 1: Functional block diagram for i^{th} module.

Table index = $|r_i - X_{i-1}|_{m_i}$ for i ≥ 2 and table content of coefficients are given as;

a_i;such that $|r_i - X_{i-1}|_{m_i} = |a_i . M_{i-1}|_{m_i}$ $0 \le a_i < m_i$

Our main goal is to keep the look up tables small and their size should also be small. This could be realized by choosing m_1 to be the largest one. Order of other moduli does not affect the memory size.

B. Implementation of three Moduli Set

Let us implement the concept of 3 moduli set for any bit word length N. The implementation holds good for all the three cases i.e. $|N|_3$= 0, $|N|_3$= 1, $|N|_3$= 2. We will discuss for $|N|_3$= 1 in detail. Figure 2 shows the conversion process, where 2 modules are required to calculate the MRC coefficients.

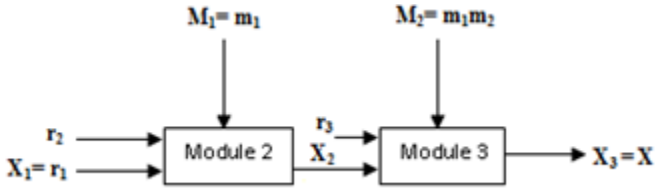


Fig.2: 3 Moduli set

Implementation.

<u>For $|N|_3$= 1</u>

According to formulae for the generation of moduli sets the moduli set is given as

$2^n - 1$, 2^{n+1}, $2^n + 1$and N = 3n + 1 (15)

Again we took N = 10. So by using equation (15), n comes out to be 3 and the moduli set should be 7, 16, 9. As is the requirement of MRC II [5], the largest modulus should be designated as m_1. Thus the moduli set are m_1 = 16, m_2 = 7 and m_3 = 9.

In our case X is represented as:

$X = X_3 = a_3 m_1 m_2 + a_2 m_1 + a_1$ (16)

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 161-173
Published Online March 2024 in IJEAST (http://www.ijeast.com)

From (14) $X_1 = a_1 = r_1$ and $M_1 = m_1$.
And

$$|r_2 - X_1|_{m_2} = \quad |a_2 . M_1|_{m_2}; \qquad (17)$$

Substituting the values of M_1 and m_2, equation (17) becomes:

$$|r_2 - X_1|_7 = |a_2.16|_7 \qquad (18)$$

In order to calculate the values of a_2 to build the look-up table, we need to know all the possible values of $|r_2 - X_1|_7$ and in our case the values are {0, 1, 2, 3, 4, 5, 6}. We do not need to know all the individual values

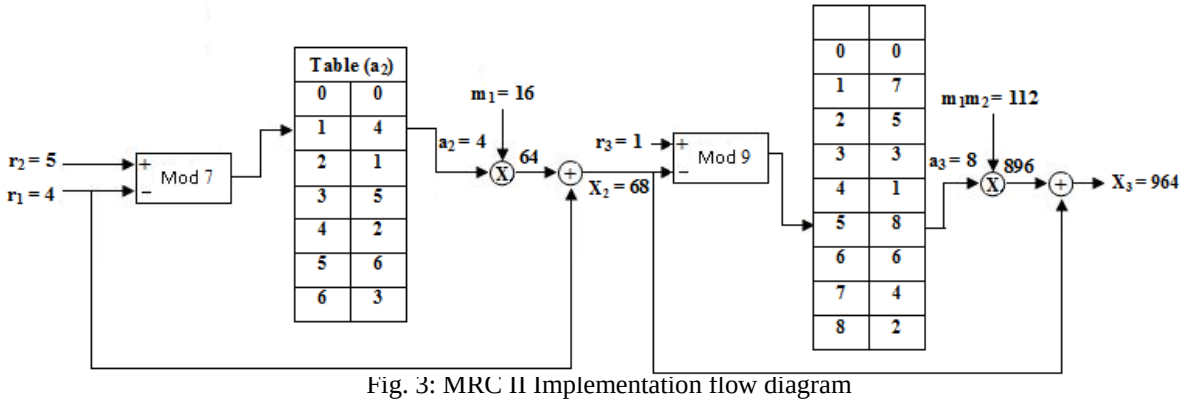


Fig. 3: MRC II Implementation flow diagram

of r_2 or X_1. Coefficient a_2 is calculated using (18)

$|r_2 - X_1|_7 = 0 = |a_2.16|_7 \Rightarrow a_2 = 0;$
$|r_2 - X_1|_7 = 1 = |a_2.16|_7 \Rightarrow a_2 = 4$

Successive values of a_2 can be calculated and hence look-up table is constructed as shown in table 1.

Table (a_2)		Table (a_3)	
$\|r_2 - X_1\|_7$	a_2	$\|r_3 - X_2\|_9$	a_3
0	0	0	0
1	4	1	7
2	1	2	5
3	5	3	3
4	2	4	1
5	6	5	8
6	3	6	6
		7	4
		8	2

Table 1: Two tables for moduli set $\{m_1, m_2, m_3\} = 16, 7, 9$

Coefficient a_3 shown above in the table are calculated in the same manner as coefficient a_2 are calculated. For the construction of table (a_3):

$$|r_3 - X_2|_{m_3} = \quad |a_3 . M_2|_{m_3} \qquad (19)$$

where $M_2 = m_1.m_2 = 16*7 = 112$ and $m_3 = 9$;

$$0 \leq r_3 < m_3 \text{ and } 0 \leq X_2 < M_2$$

Substituting the values of M_2 and m_3, equation (19) becomes

$$|r_3 - X_2|_9 = |a_3.112|_9$$

And

$|r_3 - X_2|_9 = 0 = |a_3.112|_9 \Rightarrow a_3 = 0$
$|r_3 - X_2|_9 = 1 = |a_3.112|_9 \Rightarrow a_3 = 7$

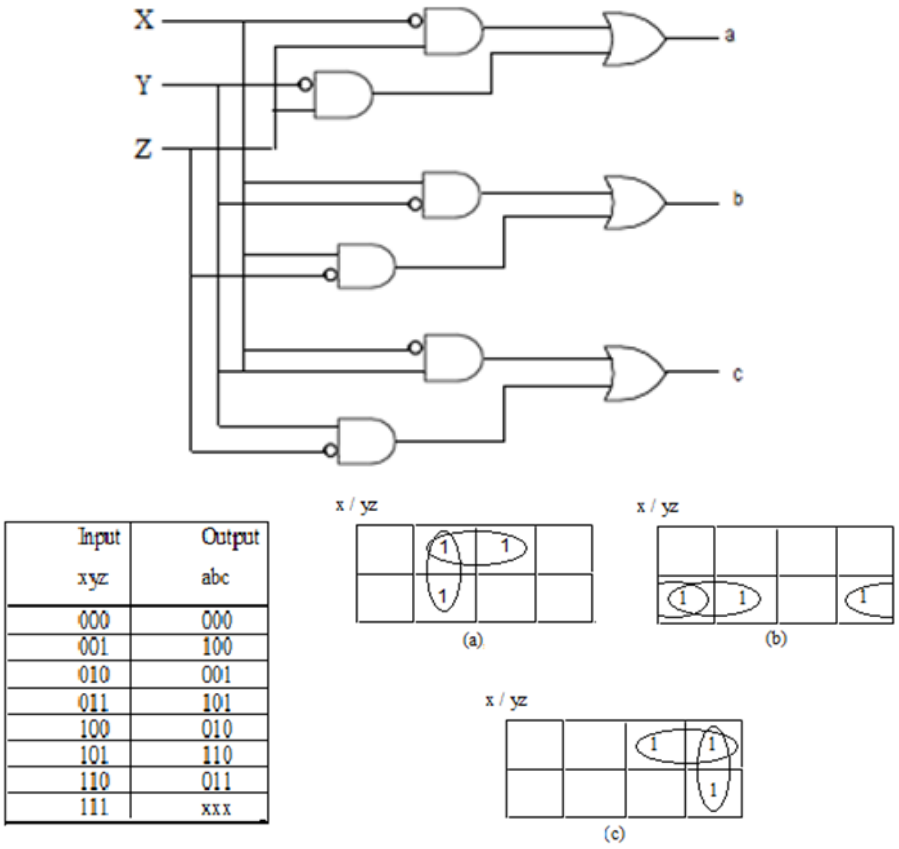
The complete flow diagram for 3 moduli set implementation is shown in figure 3.Here we have shown the conversion of RNS number (4, 5, 1) with respect to (m_1, m_2, m_3 = 16, 7, 9) to binary number 964. From look up table 1, value of $a_2 = 4$ corresponding to $|r_2 - X_1|_7 = 1$. Using formula from equation

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 161-173
Published Online March 2024 in IJEAST (http://www.ijeast.com)

(14) the value of X_2 comes out to be 68. Further the value of a_3 from look up table is 8 corresponding to $|r_3 - X_2|_7 = 5$. Again using (14) the value of X_3 is 964 as is shown as the final output in figure 3.

a. Hardware Implementation of MRC II: The look up tables used in our work is all small tables. In 3 moduli set, only two look up tables are required. Further we can see in table 1 that the values of $|r_2 - X_1|_7$ ranges from 0 to 6, so three bits (x y z) are required to represent those numbers. Similarly, coefficients of a_2 ranges from 0 to 6, so again three bits (a b c) are required to represent them as well. As shown below in look up table design xyz and abc are taken as input and output variables respectively. In case of a_3 there would be 4 input and 4 output variables.

Truth table and Circuit Implementation of a_2:



Input xyz	Output abc
000	000
001	100
010	001
011	101
100	010
101	110
110	011
111	xxx

Fig.4: Look up Table Design for a_2

The outputs of (a), (b) and (c) are:
$a = X'Z + Y'Z$ $b = X Y' + X Z'$ $c = X'Y + Y Z'$

Similarly, we can implement the other two moduli sets i.e. M_0 and M_2 for remainder zero and remainder 2 respectively. The coefficients a_2 and a_3 for the above said moduli set are calculated and shown in the table 2 and 3. Table 2 shows the coefficients for M_2 and table 3 is for M_0.

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 161-173
Published Online March 2024 in IJEAST (http://www.ijeast.com)

Table (a_2)		Table (a_3)	
$\|r_2 - X_1\|_7$	a_2	$\|r_3 - X_2\|_9$	a_3
0	0	0	0
1	2	1	8
2	4	2	7
3	6	3	6
4	1	4	5
5	3	5	4
6	5	6	3
		7	2
		8	1

Table2 : Two tables for moduli set $\{m_1, m_2, m_3\} = 32, 7, 9$

Table (a_2)		Table (a_3)	
$\|r_2 - X_1\|_8$	a_2	$\|r_3 - X_2\|_7$	a_3
0	0	0	0
1	1	1	4
2	2	2	1
3	3	3	5
4	4	4	2
5	5	5	6
6	6	6	3
7	7		

Table 3: Two tables for moduli set $\{m_1, m_2, m_3\} = 9, 8, 7$

C. RNS to Binary Converter using CRT.

In this section we will implement our three-modulus set using CRT [6]. Again, we have three cases i.e. we have one modulus set in which N is exactly divisible by 3 and the other two where we get remainder 1 or 2 when N is not exactly divisible by 3. In this implementation we will discuss one case where we get remainder 1 when N is divided by 3.

For $|N|_3 = 1$

The moduli set in this case is $M_1 = (m_1, m_2, m_3) = (2^{n+1}, 2^n\text{-}1, 2^n\text{+}1)$. Nonnegative number $X = (x_1, x_2, x_3,)$ where $X_i = |X|_{m_i}$ and i ranges from 1 to 3. The dynamic range is:

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

According to Chinese remainder theorem (CRT) number X is defined as:

$$X = \sum_{i=1}^{N} \hat{m}_i \langle \frac{1}{\hat{m}_i} \rangle_{m_i} X_i - MN(X) \qquad (21)$$

For 3 moduli set CRT equation is reduced to:

$$X = \hat{m}_1 \langle \frac{1}{\hat{m}_1} \rangle_{m_1} X_1 + \hat{m}_2 \langle \frac{1}{\hat{m}_2} \rangle_{m_2} X_2 + \hat{m}_3 \langle \frac{1}{\hat{m}_3} \rangle_{m_3} X_3 - MN(X) \qquad (22)$$

where $\hat{m}_I = \frac{M}{m_i}$; $\langle \frac{1}{\hat{m}_i} \rangle_{m_i}$ is multiplicative inverse of $\hat{m}_i$ and N(X) is an integer.

- $\hat{m}_i$ is defined as the ratio of Dynamic range M and i^{th}Moduli set [6].where i range from 1 to 3.

Consequently $\hat{m}_1 = (2^n - 1)^* (2^n + 1) = (2^{2n} - 1)$;$\hat{m}_2 = (2^{n+1})^* (2^n + 1)$' and$\hat{m}_3 = (2^{n+1})^* (2^n - 1)$ (23)

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 161-173
Published Online March 2024 in IJEAST (http://www.ijeast.com)

- **Multiplicative inverses** $\langle \frac{1}{\hat{m}_i} \rangle_{m_i}$ are given in the following theorem:

Theorem 2: For the moduli set $(m_1, m_2, m_3) = (2^{n+1}, 2^n - 1, 2^n + 1)$ the multiplicative inverses are:

$\langle \frac{1}{\hat{m}_1} \rangle_{m_1} = \langle -1 \rangle_{m_1}$

$\langle \frac{1}{\hat{m}_2} \rangle_{m_2} = \langle 2^{n-2} \rangle_{m_2}$

And $\langle \frac{1}{\hat{m}_3} \rangle_{m_3} = \langle -2^{n-2} \rangle_{m_3}$

Proof: We will show that for i = (1, 2, 3) $\langle \langle \hat{m}_i \rangle_{m_i} \langle \frac{1}{\hat{m}_i} \rangle_{m_i} \rangle_{m_i} = 1$

For i = 1; $\langle \hat{m}_1 \rangle_{m_1} = \langle 2^{2n} - 1 \rangle_{2^{n+1}} = \langle -1 \rangle_{2^{n+1}}$

$$\langle \frac{1}{\hat{m}_1} \rangle_{m_1} = \langle -1 \rangle_{2^{n+1}} \quad (24)$$

$$\langle \langle \hat{m}_1 \rangle_{m_1} \langle \frac{1}{\hat{m}_1} \rangle_{m_1} \rangle_{m_1} = \langle (-1)(-1) \rangle_{2^{n+1}} = 1$$

Fori = 2

$$\langle \hat{m}_2 \rangle_{m_2} = \langle (2^{n+1})(2^n + 1) \rangle_{2^n - 1} = \langle 4 \rangle_{2^n - 1}$$

$$\langle \frac{1}{\hat{m}_2} \rangle_{m_2} = \langle 2^{n-2} \rangle_{2^n - 1} \quad (25)$$

$$\langle \langle \hat{m}_2 \rangle_{m_2} \langle \frac{1}{\hat{m}_2} \rangle_{m_2} \rangle_{m_2} = \langle (4)(2^{n-2}) \rangle_{2^n - 1}$$
$$= \langle (2^n) \rangle_{2^n - 1} = \langle (2^n + 1 - 1) \rangle_{2^n - 1} = 1$$

For i = 3

$$\langle \hat{m}_3 \rangle_{m_3} = \langle (2^{n+1})(2^n - 1) \rangle_{2^n + 1} = \langle 4 \rangle_{2^n + 1}$$

$$\langle \frac{1}{\hat{m}_3} \rangle_{m_3} = \langle -2^{n-2} \rangle_{2^n + 1} \quad (26)$$

$$\langle \langle \hat{m}_3 \rangle_{m_3} \langle \frac{1}{\hat{m}_3} \rangle_{m_3} \rangle_{m_3} = \langle (4)(-2^{n-2}) \rangle_{2^n + 1}$$
$$= \langle (-2^n) \rangle_{2^n + 1} = \langle (2^n + 1 - 2^n) \rangle_{2^n + 1} = 1$$

We have all the parameters that are required in equation (22).Substituting we get

$$X = -(2^{2n} - 1)X_1 + [(2^{n+1})(2^n + 1)(2^{n-2})]X_2 + [(2^{n+1})(2^n - 1)(-2^{n-2})] - (2^{n+1})(2^n - 1)(2^n + 1)N(X) \quad (27)$$

Divide equation (27) by 2^{n+1} and take the floor value of both sides.

$$\left\lfloor \frac{X}{2^{n+1}} \right\rfloor = -\lfloor 2^{n-1} \rfloor X_1 + \lfloor 2^{2n-2} + 2^{n-2} \rfloor X_2 + \lfloor -2^{2n-2} + 2^{n-2} \rfloor X_3 - (2^{2n} - 1)N(X) \quad (28)$$

In order to eliminate the term $(2^{2n} - 1)N(X)$ take modulus $2^{2n} - 1$ both sides of (28)

$$\left\lfloor \frac{X}{2^{n+1}} \right\rfloor = \langle -2^{n-1} X_1 \rangle_{2^{2n}-1} + \langle (2^{2n-2} + 2^{n-2}) X_2 \rangle_{2^{2n}-1} + \langle (-2^{2n-2} + 2^{n-2}) X_3 \rangle_{2^{2n}-1} \quad (29)$$

X is represented by equation:

$$X = \left\lfloor \frac{X}{2^{n+1}} \right\rfloor 2^{n+1} + X_1 \quad (30)$$

Equation (29) is fractionated in parts as u_1, u_2 and u_3 and is expressed as:

$$\left\lfloor \frac{X}{2^{n+1}} \right\rfloor = \langle u_1 + u_2 + u_3 \rangle_{2^{2n}-1} \quad (31)$$

where $u_1 = \langle -2^{n-1} X_1 \rangle_{2^{2n}-1}$;
$u_2 = \langle (2^{2n-2} + 2^{n-2}) X_2 \rangle_{2^{2n}-1}$
$u_3 = \langle (-2^{2n-2} + 2^{n-2}) X_3 \rangle_{2^{2n}-1}$

In order to realize (31) we need to obtain the simplified expression of $u_1, u_2 \, and \, u_3$ by taking modulus of respective terms.

Evaluation of $u_1, u_2 \, and \, u_3$: For the Binary representation of X_i, x_{ij} represents j^{th} bit and its weight is expressed as 2^j. The complement is given as $\bar{x}_{ij}$.

- **Evaluating u_1:** X_1 is n + 1 bit and its 2n bit representation is:

$$X_1 = \overbrace{0\,0 \ldots 0\,0}^{n-1} \overbrace{x_{1n}\, x_{1(n-1)} \ldots x_{11} x_{10}}^{n+1}$$

$$-X_1 = \overbrace{1\,1 \ldots 1\,1}^{n-1} \overbrace{\bar{x}_{1n}\, \bar{x}_{1(n-1)} \ldots \bar{x}_{11} \bar{x}_{10}}^{n+1}$$

For u_1 multiply above equation with 2^{n-1} and take modulus $2^{2n} - 1$, so that equation looks like:

$$u_1 = \langle -2^{n-1} X_1 \rangle_{2^{2n}-1} =$$

$$\overbrace{\bar{x}_{1n}\, \bar{x}_{1(n-1)} \ldots \bar{x}_{11} \bar{x}_{10}}^{n+1} \overbrace{1\,1 \ldots 1\,1}^{n-1} \quad (32)$$

- **Evaluating u_2:** X_2 is n bit. For simplification we split u_2 into two parts;

$u'_2 = \langle 2^{2n-2} X_2 \rangle_{2^{2n}-1}$ and $u''_2 = \langle 2^{n-2} X_2 \rangle_{2^{2n}-1}$ 2n bit representation of X_2 is:

$$X_2 = \overbrace{0\,0 \ldots 0\,0}^{} \overbrace{x_{2(n-1)}\, x_{2(n-2)} \ldots x_{21} x_{20}}^{}$$

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 161-173
Published Online March 2024 in IJEAST (http://www.ijeast.com)

$\boldsymbol{u'_2} = \langle 2^{2n-2} X_2 \rangle_{2^{2n}-1} =$

$$\overbrace{x_{21}x_{20}\,0\,0\,.\,.\,.\,.}^{n}\;\overbrace{0\,0\,x_{2(n-1)}x_{2(n-2)}\,.\,.\,.\,.}^{n} \quad (33)$$

$\boldsymbol{u''_2} = \langle 2^{n-2} X_2 \rangle_{2^{2n}-1} =$

$$\overbrace{0\,0\;x_{2(n-1)}x_{2(n-2)}\,.\,.\,.\,.}^{n}\;\overbrace{x_{21}x_{20}\,0\,0\,.\,.\,.}^{n} \quad (34)$$

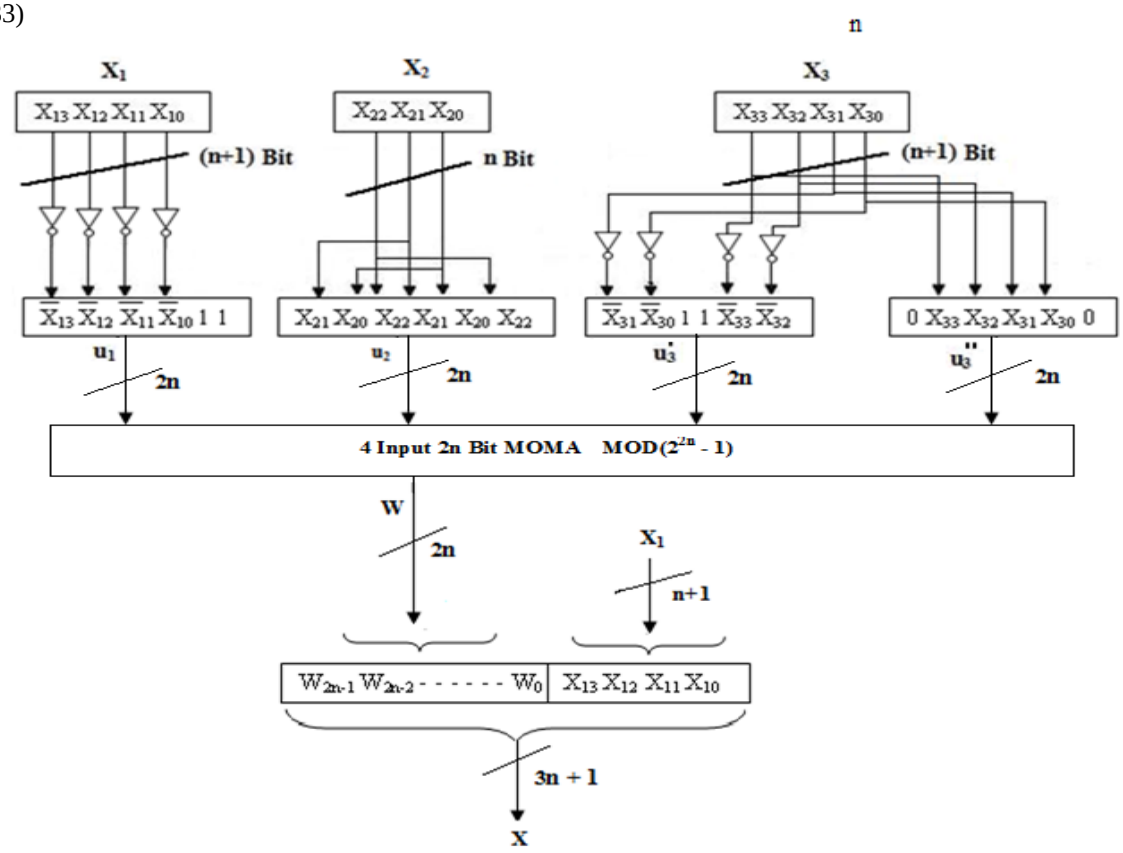


Fig. 5: Functional Block Diagram

$u_2 = \boldsymbol{u'_2} + \boldsymbol{u''_2} = \langle (2^{2n-2} + 2^{n-2}) X_2 \rangle_{2^{2n}-1}$

$$u_2 = \overbrace{x_{21}x_{20}x_{2(n-1)}x_{2(n-2)}\,.\,.\,.\,.}^{n}\;\overbrace{x_{21}x_{20}x_{2(n-1)}x_{2(n-2)}\,.\,.\,.\,.}^{n} \quad (35)$$

- **Evaluating $\boldsymbol{u_3}$:** X_3 is n+1 bit. Again, for simplification we split u_3 in two parts;

$\boldsymbol{u'_3} = \langle -2^{2n-2} X_3 \rangle_{2^{2n}-1}$ and $\boldsymbol{u''_3} = \langle 2^{n-2} X_3 \rangle_{2^{2n}-1}$
2n bit representation of X_3 is:

$$X_3 = \overbrace{0\,0\,.\,.\,.\,.\,0\,0}^{n-1}\overbrace{x_{3n}x_{3(n-1)}\,.\,.\,.\,x_{31}x_{30}}^{n+1}$$

$$-X_3 = \overbrace{1\,1\,.\,.\,.\,1\,1}^{n-1}\overbrace{\bar{x}_{3n}\bar{x}_{3(n-1)}\,.\,.\,.\,\bar{x}_{31}\bar{x}_{30}}^{n+1}$$

$\boldsymbol{u'_3} = \langle -2^{2n-2} X_3 \rangle_{2^{2n}-1} =$

$$\overbrace{\bar{x}_{31}\bar{x}_{30}\,1\,1\,.\,.\,.}\overbrace{1\,1\,\bar{x}_{3n}\bar{x}_{3(n-1)}\,.\,.\,.} \quad (36)$$

$\boldsymbol{u''_3} = \langle 2^{n-2} X_3 \rangle_{2^{2n}-1} =$

$$\overbrace{0\,x_{3n}x_{3(n-1)}.\,.\,.\,.}\overbrace{x_{32}x_{31}x_{30}\,0\,0\,.\,.\,.\,0} \quad (37)$$

$u_3 = \boldsymbol{u'_3} + \boldsymbol{u''_3} = \langle (-2^{2n-2} + 2^{n-2}) X_3 \rangle_{2^{2n}-1}$

In u_3, both $\boldsymbol{u'_3}$ and $\boldsymbol{u''_3}$ are 2n bits and are overlapping. Thus we took $\boldsymbol{u'_3}$ and $\boldsymbol{u''_3}$ separately. Now we have simplified u_1, u_2, u_3', u_3'' so that these parameters could be incorporated in equation (31) for the realization. Eventually we need to get the value of X from equation (30). The parameter $\left\lfloor \frac{X}{2^{n+1}} \right\rfloor$ is multiplied by 2^{n+1} or a shifting $\left\lfloor \frac{X}{2^{n+1}} \right\rfloor$ by n + 1 bits and adding X_1, an n + 1 bit number. The complete functional block

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 161-173
Published Online March 2024 in IJEAST (http://www.ijeast.com)

Table 4: Expression for U_1, U_2 and U_3

$\|N\|_3 = 0$	$\|N\|_3 = 1$	$\|N\|_3 = 2$
$u_1 = \langle -2^n X_1 \rangle_{2^{2n}-1}$	$u_1 = \langle -2^{n-1} X_1 \rangle_{2^{2n}-1}$	$u_1 = \langle -2^{n-2} X_1 \rangle_{2^{2n}-1}$
$u_2 = \langle (2^{2n-1} + 2^{n-1}) X_2 \rangle_{2^{2n}-1}$	$u_2 = \langle (2^{2n-2} + 2^{n-2}) X_2 \rangle_{2^{2n}-1}$	$u_2 = \langle (2^{2n-3} + 2^{n-3}) X_2 \rangle_{2^{2n}-1}$
$u_3 = \langle (-2^{2n-1} + 2^{n-1}) X_3 \rangle_{2^{2n}-1}$	$u_3 = \langle (-2^{2n-2} + 2^{n-2}) X_3 \rangle_{2^{2n}-1}$	$u_3 = \langle (-2^{2n-3} + 2^{n-3}) X_3 \rangle_{2^{2n}-1}$

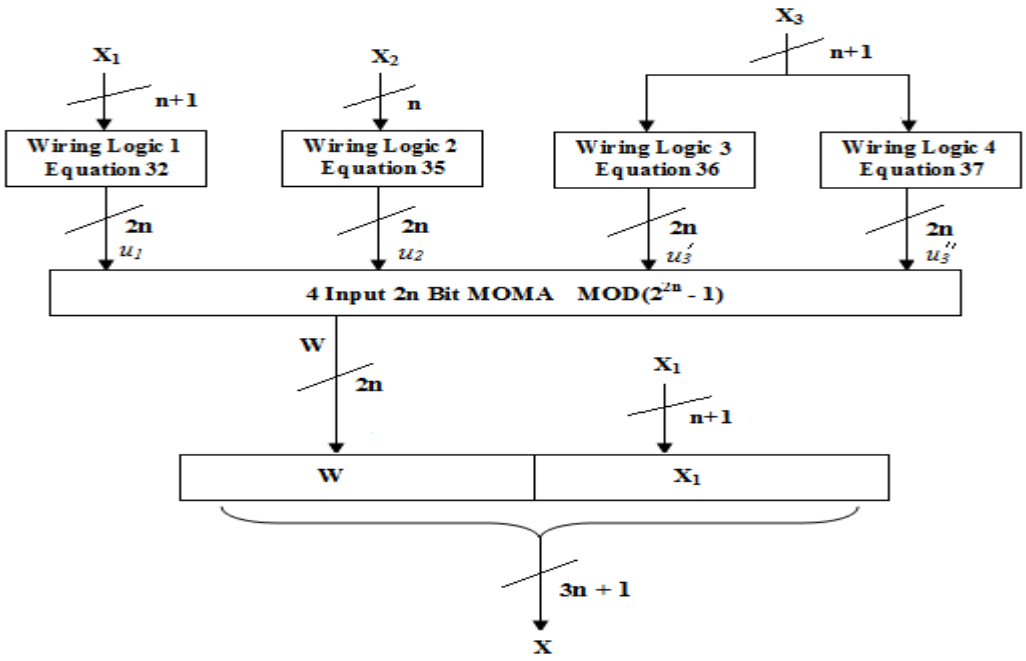


Fig. 6: Implementation of 3 modulii set for M_1

diagram is shown below in figure 5, where W = $\left\lfloor \frac{X}{2^{n+1}} \right\rfloor$. The computation of X is achieved by wiring operations, eliminating the need for multiplication and addition operations.

On similar lines we can drive expressions for u_1, u_2 and u_3 for $|N|_3 = 0$ and $|N|_3 = 2$. Table 4 shows u_1, u_2 and u_3 for $|N|_3 = 0$, $|N|_3 = 1$ and $|N|_3 = 2$.

The following example elaborates the case for N = 10; i = $|10|_3$ = 1 and n = 3.

Example 4.1: For the Modulus set (16, 7, 9), convert the residue representation (4, 5, 1) into its binary equivalent.
From equation (31) we need to calculate u_1, u_2 and u_3.
$X_1 = 4$; $X_1 = X_{13}X_{12}X_{11}X_{10} = 0\ 1\ 0\ 0$, thus from equation (32), the value of u_1 is given as:

$u_1 = \bar{X}_{13}\bar{X}_{12}\bar{X}_{11}\bar{X}_{10} | 1\ 1 = 1\ 0\ 1\ 1\ 1\ 1 = 47$

$X_2 = 5$; $X_2 = X_{22}X_{21}X_{20} = 1\ 0\ 1$, from equation (35), the value of u_2 is given as:

$u_2 = X_{21}X_{20}X_{22} | X_{21}X_{20}X_{22} = 0\ 1\ 1\ 0\ 1\ 1 = 27$

$X_3 = 1$; $X_3 = X_{33}X_{32}X_{31}X_{30} = 0\ 0\ 0\ 1$, from equation (36), (37), the value of u_3' and u_3'' is:

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 161-173
Published Online March 2024 in IJEAST (http://www.ijeast.com)

Table 5: Binary form values of U_1, U_2 and U_3 for $|N|_3 = 0$, $|N|_3 = 1$ and $|N|_3 = 2$

Case / **Notation**	$\|N\|_3 = 0$	$\|N\|_3 = 1$	$\|N\|_3 = 2$
For U_1	X_1 is n bit $X_1 = x_{1(n-1)}x_{1(n-2)} \dots x_{11}x_{10}$ $U_1 = \bar{x}_{1(n-1)}\bar{x}_{1(n-2)} \dots \bar{x}_{11}\bar{x}_{10}$ $\|1\ 1\ \dots\ 1\ 1$	X_1 is n +1 bit $X_1 = x_{1n}x_{1(n-1)} \dots x_{11}x_{10}$ $U_1 = \bar{x}_{1n}\bar{x}_{1(n-1)} \dots \bar{x}_{11}\bar{x}_{10}$ $\|1\ 1\ \dots\ 1\ 1$	X_1 is n +2 bit $X_1 = x_{1(n+1)}x_{1n} \dots x_{11}x_{10}$ $U_1 = \bar{x}_{1(n+1)}\bar{x}_{1n} \dots \bar{x}_{11}\bar{x}_{10}$ $\|1\ 1\ \dots\ 1\ 1$
For U_2	X_2 is n bit $X_2 = x_{2(n-1)}x_{2(n-2)} \dots x_{21}x_{20}$ $U_2 = x_{20}x_{2(n-1)}x_{2(n-2)} \dots x_{21}$ $\|x_{20}x_{2(n-1)}x_{2(n-2)} \dots x_{21}$	X_2 is n bit $X_2 = x_{2(n-1)}x_{2(n-2)} \dots x_{21}x_{20}$ $U_2 = x_{21}x_{20}x_{2(n-1)}x_{2(n-2)} \dots$ $\|x_{21}x_{20}x_{2(n-1)}x_{2(n-2)} \dots$	X_2 is n bit $X_2 = x_{2(n-1)} \dots x_{22}x_{21}x_{20}$ $U_2 = x_{22}x_{21}x_{20}x_{2(n-1)} \dots$ $\|x_{22}x_{21}x_{20}x_{2(n-1)} \dots$
For U_3 **$U_3' + U_3''$**	X_3 is n + 1 bit $X_3 = x_{3n}x_{3(n-1)} \dots x_{31}x_{30}$ $U_3' = \bar{x}_{30}1\ 1 \dots 1\ \|$ $1\ \bar{x}_{3n}\bar{x}_{3(n-1)} \dots \bar{x}_{31}$ $U_3'' = x_{3n}x_{3(n-1)} \dots x_{31}\|x_{30}$ $0\ 0 \dots 0\ 0$	X_3 is n + 1 bit $X_3 = x_{3n}x_{3(n-1)} \dots x_{31}x_{30}$ $U_3' = \bar{x}_{31}\bar{x}_{30}11 \dots\|$ $1\ 1\ \bar{x}_{3n}\bar{x}_{3(n-1)} \dots$ $U_3'' = 0\ x_{3n}\ x_{3(n-1)} \dots \|x_{32}x_{31}$ $x_{30}\ 0\ 0\ .\ .\ .\ 0$	X_3 is n + 1 bit $X_3 = x_{3n} \dots x_{32}x_{31}x_{30}$ $U_3' = \bar{x}_{32}\bar{x}_{31}\bar{x}_{30}1 \dots \|$ $1\ 1\ 1\ \bar{x}_{3n}\ \dots$ $U_3'' = 0\ 0\ x_{3n}\ \dots\|$ $x_{33}x_{32}x_{31}x_{30}0 \dots 0$

Table 6: Time Delay and Hardware Complexity Comparison

Converter	**Moduli Set***	**FA**	**HA**	**AND/OR**	**XOR/NOR**	**Other**	**Delay**
[19]	M_3	6n – 1	3n – 7	----	----	(n-1)MUX	$(4n)t_{FA}$
[15]	M_0	6n + 1	----	n + 3	n + 1	2n MUX	$(2n+2)t_{FA} + t_{MUX}$
[17,18]	M_0	4n	----	2	----	----	$(4n + 1)t_{FA}$
[16] CI	M_0	4n	1	----	1	2 MUX	$(4n + 1)t_{FA}$
[16] CII	M_0	6n	1	1	1	(2n+2) MUX	$(n + 1)t_{FA}$
[16] CIII	M_0	4n	1	2n + 2	2n – 1	(2n+2) MUX	$(n + 1)t_{FA}$
[14] CI	M_4	4n + 3	----	N	n	----	$(6n + 5)t_{FA}$
[14] CII	M_4	14n + 21	2n + 3	----	----	(2n+1)3:1MUX	$(2n + 7)t_{FA}$
[14] CIII	M_4	12n + 19	2n + 2	----	----	$10(2n+1)A_{ROM}$ (2n+2)2:1MUX	$(2n + 7)t_{FA}$
Ours	M_0, M_1, M_2	6n	----	----	----	2n + 1 + i Inverters	$(2n+2)t_{FA}$

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 161-173
Published Online March 2024 in IJEAST (http://www.ijeast.com)

***** $M_0 = \{2^n - 1, 2^n, 2^n + 1\}$, $M_1 = \{2^n - 1, 2^{n+1}, 2^n + 1\}$, $M_2 = \{2^n - 1, 2^{n+2}, 2^n + 1\}$, $M_3 = \{2^n - 1, 2^n, 2^{n-1} - 1\}$ and $M_4 = \{2^{n+1} - 1, 2^n, 2^n - 1\}$.

Table 7: Comparison of various Converters for Three Moduli set in terms of Dynamic Range Efficiency (Deff)

Word Length (N)	**Value of n**	**Mouldi Set**	**$D_{eff} = M/2^N$**
$\|N\|_3 = 0$	N = 3n	M_0	99%
		M_1	200%
		M_2	393%
		M_3	33%
		M_4	164%
$\|N\|_3 = 1$	N = 3n + 1	M_0	50%
		M_1	99%
		M_2	197%
		M_3	17%
		M_4	82%
$\|N\|_3 = 2$	N = 3n + 2	M_0	25%
		M_1	50%
		M_2	99%
		M_3	8%
		M_4	41%

$u_3' = \overline{X}_{31}\overline{X}_{30} | 1\ 1\ \overline{X}_{33}\overline{X}_{32} = 1\ 0\ 1\ 1\ 1\ 1$

And $u_3'' = 0\ X_{33} | X_{32}X_{31}X_{30}0 = 0\ 0\ 0\ 0\ 1\ 0$

Again, from the assumption that $\left\lfloor \frac{X}{2^{n+1}} \right\rfloor$ = W and following equation (31),

$\left\lfloor \frac{X}{2^{n+1}} \right\rfloor = W = \langle u_1 + u_2 + u_3 \rangle_{2^{2n}-1}$

$W = \langle 47 + 27 + 49 \rangle_{63} = 1\ 1\ 1\ 1\ 0\ 0 = 60$

From equation (30) $X = W*2^{n+1} + X_1 = \overbrace{1\ 1\ 1\ 1\ 0}^{W}\overbrace{0\ 0\ 1\ 0\ 0}^{X_1} =$ 964 (39)

Here we can see that X is obtained by shifting W, n + 1 bit to the left and appending X_1. Figure 6 shows the implementation of this example.

Theorem 3: For any word length N, the three moduli set is $\{2^{n+i}, 2^n - 1, 2^n + 1\} = \{m_1, m_2, m_3\}$, where $i = |N|_3$. Any integer X in [0, M – 1] has RNS representation of (x_1, x_2, x_3), where $X_j = |X|_{m_j}$ and the residue size is (n + i, n, n + 1) respectively. The Value of X can be computed from
$X = W*2^{n+i} + X_1$

Where $W = \langle u_1 + u_2 + u_3 \rangle_{2^{2n}-1}$ and $u_j = f\langle x_j \rangle$ in 2n bit representation. The value of u_j is summarized in table 5 for each i. The size of X is 3n + i.

Hardware Architecture:
Theorem 3 states that regardless of i, the hardware architecture is the same. It consists of rewiring each residue x_j to compute u_j in 2n bit representation. The u_j's are then added mod $(2^{2n} - 1)$ to compute W. The final answer X is computed by shifting W by n + i bit and appending X_1 of size n + i bit, i.e. no overlapped. The hardware architecture is shown in figure 5 regardless of i. The only hardware to be implemented is the mod $(2^{2n} - 1)$ adder, which is shown in figure 7. The propagation delay and hardware count are determined by the mod $(2^{2n} - 1)$ adder. The wiring operations only require inverters and introduce one inverter delay. The number of inverters required is determined when computing u_j. u_1 needs to compute $-X_1$, which requires n + 1 (size of X_1) inverters; u_2 does not require any inverter and u_3 needs to compute $-X_3$ which requires n + 1 (size of X_3) inverters i.e. total number of inverters is n + 1 + i, where $i = |N|_3$.
The mod adder architecture consists of 2 CSA (2n) and each CSA (2n) contribute 1 FA delay. The 2 operands mod $(2^{2n} - 1)$ adder is implemented using RCA (2n) which contributes 2n

International Journal of Engineering Applied Sciences and Technology, 2024
Vol. 8, Issue 11, ISSN No. 2455-2143, Pages 161-173
Published Online March 2024 in IJEAST (http://www.ijeast.com)

FA delay. The 1 inverter delay is insignificant compared to FA delay and thus the total delay is 2n + 2. Each CSA (2n) has 2n FA and RCA (2n) has 2n FA, therefore a total of 6n FAs are used. The mod adder architecture is shown in figure 7.

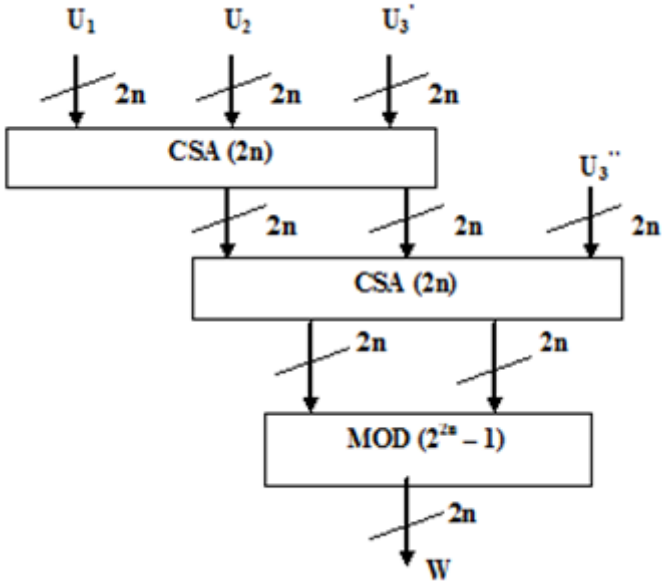


Fig. 7: Architecture of mod $2^{2n} - 1$ adder

V. COMPARISION

In table 6, the comparison is done in terms of delay time and table 7 did the comparison in terms of dynamic range [20-23].Looking at table 6 we can say that in our case we are using only 6n full adders and 2n +1 + i inverters, whereas all other converters require the use of one or more components such as gates (OR, NAND, XOR, NOR), Half Adders and Multiplexers in addition to Full adders, thus reducing the hardware complexity to its minimum threshold value. If we look at the delay time, we are superior to all the converters except CII, CIII [16].

In table 7, we compare the Dynamic Range Efficiency (D_{eff}) of the 5 Moduli sets with respect to word length N. The best D_{eff} of each moduli set is as follows:

M_0 D_{eff} = 99% if $|N|_3 = 0$
M_1 D_{eff} = 99% if $|N|_3 = 1$
M_2 D_{eff} = 99% if $|N|_3 = 2$
M_3 D_{eff} = 33% if $|N|_3 = 0$
M_4 D_{eff} = 82% if $|N|_3 = 1$

Thus, we can say that our proposed moduli set M_0, M_1 and M_2 are the most efficient as far as D_{eff} is concerned.

The proposed Moduli set M_0, M_1 and M_2 have the following properties.

1. Excellent Propagation Delay.
2. Simple Hardware Architecture.
3. The Best D_{eff}.

VI. CONCLUSION

In this paper we proposed a new three moduli set (2^n -1, 2^{n+i}, 2^n +1), where i =$|N|_3$. It is shown to have the best Dynamic Range Efficiency for any N. The CRT implementation has the same architecture for any i. It consists of wiring operations and one mod adder operations. The Hardware complexity is 6n FA, 2n + 1 + i inverters with competitive propagation delay of $(2n + 2)t_{FA.}$

VII. REFERENCES

[1]. N.S. Szabo, R.I. Tanaka, Residue Arithmetic and its Application to Computer Technology, McGraw-Hill, New York, 1967.

[2]. [2] M. Soderstrand, W. Jenkins, G. Jullien, F. Taylor (Eds.), Residue Number System Arithmetic, Modern Applications in Digital Signal Processing, IEEE Press, New York, 1986.

[3]. W.K. Jenkins and B.J. Leon, The use of Residue Number Systems in Design of finite impulse response digital filters, IEEE Trans. Circuits syst. Vol. CSA-24, no. 4 pp 191-201. April 1977.

[4]. W. Wang, M.N.S. Swamy, M. Ahmad and Y. Wang, A Comprehensive Study of three Moduli Sets for Residue Arithmetic, Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering, Edmonton, Alberta, Canada May 9-12 1999.

[5]. M.Akkal and P. Siy, A new Mixed Radix Conversion Algorithm MRC-II, Journal of System Architecture 53 (2007) 577-586.

[6]. A.Hiasat and A Sweidan, Residue Number System to Binary Converter for the Moduli set (2^{n-1},2^n – 1,2^n + 1), Journal of System Architecture 49 (2003) 53-58.

[7]. A.Benjamin Premkumar, An RNS to Binary Converter in 2n + 1,2n, 2n – 1Moduli set, IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, vol. 39, no. 7, July 1992.

[8]. A.Benjamin Premkumar, A RNS to Binary in a three moduli set with common factors, IEEE Transactions on Circuits and systems – II: Analog and Digital Signal Processing, vol. 42, no. 4, April 1995.

[9]. Ahmad A. Hiasat and Hoda S. Abdel-Aty-ZohdyResidue-to-Binary Arithmetic Converter forthe Moduli Set (2^k, 2^k-1, 2^{k-1}- 1), IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing, vol. 45, no. 2, February 1998.

[10]. S.H. Lin, M.H. Sheu, J.S.Lin, W.T Sheu, Efficient VLSI Design for RNS Reverse Converter Based on New Moduli Set (2^n – 1, 2^n + 1, 2^{2n+1}), IEEE 2006, pp. 2020-2023.

[11]. Yuke Wang, Residue-to-Binary Converters Based on NewChinese Remainder Theorems, IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing, vol. 47, no. 3, March 2000.

[12]. A.SabbaghMolahosseini, M. Kuchaki Rafsanjani, S.H. Ghafouri, M. Hashemipour, A Reduced-Area Reverse Converter for the Moduli Set (2^n, 2^n-1, 2^{2n-1}– 1), International Journal of Advancements in Computing Technology Volume 2, Number 5, December 2010.

[13]. Arash Hariri, KeivanNavi, Reza Rastegar, A new High Dynamic Range Moduli set with Efficient Reverse Converter, Computers and Mathematics with Applications 55 (2008) 660–668.

[14]. Pemmaraju V. Ananda Mohan,RNS-To-Binary Converter for a New Three-Moduli set {2^{n+1}– 1, 2^n, 2^n −1,}, IEEE Transactions on Circuits and Systems – II: Express Briefs, vol. 54, no. 9, September 2007

[15]. M. Bharadwaj, A. B.Premkumar, and T. Srikanthan, Breaking the2n-bit carry propagation barrier in residue to binary conversion forthe (2^n – 1, 2^n, 2^n + 1) moduli set, IEEE Trans. Circuits Syst. I,Fundam. Theory Appl., vol. 45, no. 9, pp. 998–1002, Sep. 1998.

[16]. Y.Wang, X. Song, M. Aboulhamid, and H. Shen, Adder-based residue to binary number converters for (2^n – 1, 2^n, 2^n + 1), IEEE Trans.Signal Processing, vol. 50, no. 7, pp. 1772–1779, Jul. 2002.

[17]. S. J. Piestrak, A high-speed realization of residue to binary system converter, IEEE Trans. Circuits Syst. II, Analog Digit. Signal Processing, vol. 42, no. 10, pp. 661–663, Oct. 1995.

[18]. A.Dhurkadas, Comments on 'A high-speed realization of a residueto binary number system converter', IEEE Trans. Circuits Syst. II, Analog Digit. Signal Processing, vol. 45, no. 3, pp. 446–447, Mar. 1998

[19]. W. Wang, M.N.S. Swamy,M.O. Ahmad, and Y.Wang, Ahigh-speedresidue-to-binary converter for three-moduli (2^k, 2^k-1, 2^{k-1}- 1)RNS and a scheme for its VLSI implementation, IEEE Trans. CircuitsSyst. II, Analog Digit. Signal Processing, vol. 47, no. 12, pp. 1576–1581, Dec. 2000.

[20]. Tkatchenko, Tatiana V., Yimin Shen, Rod D. Braun, Gurinder Bawa, Pradeep Kumar, Ivan Avrutsky, and Andrei V. Tkatchenko. "Photopic visual input is necessary for emmetropization in mice." Experimental eye research 115 (2013): 87-95.

[21]. Bawa, Gurinder, Tatiana V. Tkatchenko, Ivan Avrutsky, and Andrei V. Tkatchenko. "Variational analysis of the mouse and rat eye optical parameters." Biomedical optics express 4, no. 11 (2013): 2585-2595.

[22]. Bawa, Gurinder. Modeling of mouse eye and errors in ocular parameters affecting refractive state. Wayne State University, 2013.

[23]. Dajani, O., Bawa, G., & Singh, H. (2012). VLSI Implementation of Residue Adder and Subtractor. In Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS) (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).