# A High-Speed Residue-to-Binary Converter for Three-Moduli $\{2^{2n+2}-1, 2^{2n+1}-1, 2^{n}\}$ set

**Amani Goniemat, Andraws Swidan**
**Computer Engineering Department, The University of Jordan**
**Amman, Jordan**

**Abstract**
Residue-to-binary conversion is the crucial step for residue arithmetic. The traditional methods are the Chinese Remainder Theorem and the Mixed Radix Conversion. Both approaches have some well known long standing difficulties, new Chinese remainder theorem used to overcome those difficulties. In this paper presents, and a new converter for specific moduli set $\{2^{2n+2}-1, 2^{2n+1}-1, 2^{n}\}$ was proposed. Our proposed converter is based on the CRTI. A detailed comparative analysis of the proposed converter was carried out. The analysis showed that our proposed converter overcomes other converters by speed and at the same time it is comparable to them by hardware requirements.
***Keywords:*** Residue Number System, Reverse Converters, New Chinese Remainder Theorem.

## 1. Introduction

Residue number system (RNS) is a non-weighted system and uses residues of a number in particular modulus for its representation.

Arithmetic operations on residues can be performed in each moduli in parallel without carry propagation between them, thus one of the important characteristic of using residue arithmetic is the carry-free property which increases the calculation speed and decrease the consumed power.

In addition instead of performing arithmetic operations on large number, calculations are done on its corresponding residues in parallel. Hence the hardware requirement is reduced and speed operations is improved moreover all tasks are performed parallel.

Considering the characteristics of RNS, it has been applied on many arithmetic operations such as fast number theoretic transforms, discrete Fourier transforms and many other areas. Also it received a considerable attention since 1950 in image processing, digital filters and digital signal processing computation algorithms (DSP).

Unlike conventional number system RNS bear the extra cost of conversion step that is user to interface the RNS with the external world either for convert the binary to residue representation for forward conversion or the inverse in reverse conversion to produce the binary equivalent of residues. Other critical issue concerning the use of RNS is the choice of moduli set as the form of the moduli set and the number of moduli that chosen for RNS processor affects on dynamic range, speed and its VLSI implementation.

Up to now, many moduli sets have been presented with various dynamic ranges either 3n, 4n, 5n bits with three, four, five and even six moduli sets but always the trend is to offer a moduli set that meet high performance needs with a large dynamic range and parallelism.

In this paper a new three modules $\{2^{2n+2}-1, 2^{2n+1}-1, 2^{n}\}$ with (5n)-bits dynamic range is proposed and a high speed and low complex reverse converter is designed based on new CRT algorithm.

## 2. Related background

The Residue Number System is defined in terms of a relatively-prime modulus set $\{P_{1,} P_{1,} \ldots., P_n\}$ where gcd $(P_i, P_j) = 1$ for $i \neq j$ , and gcd (a, b) denotes the greatest common divisor of a and b. A weighted number X can be represented as $X = (x_1, x_2, \ldots., x_n)$, where this representation is unique for any integer X in range$[0, M-1]$, where M is the Dynamic range of the modulus set and defined as $M = P_1 P_2 \ldots \ldots P_n$.

In order to convert from binary to residue numbers and vice-versa, a binary to residue (forward converter) is required in the front end of the system and a residue to binary in the bank end of the system. Reverse conversion involves a significant degree of complexity; hence an efficient design of reverse converter greatly simplifies the operations in RNS.

The algorithms of residue to binary conversion are mainly based on the Chinese remainder theorem (CRT) and mixed radix conversion (MRC), and recently a new implementation of (CRT) is proposed defined as (CRT-I) and (CRT-I I). The New CRTs have potentiality to create higher performance reverse converters than CRT and MRC particularly for some special moduli sets. Hence, many researchers have been done in the recent years to discover efficient moduli sets which can be fitted with properties of New CRTs.

Chinese Remainder Theorem: Let $m_1, m_2, \ldots, m_n$ be pairwise relatively prime positive integers, i.e. $\gcd(m_i, m_j) = 1$ for $i \neq j$.. The system

$$x \equiv a_1 (mod m_1)$$
$$x \equiv a_2 \quad (mod m_2)$$

.
.
.

$$x \equiv a_n \quad (mod m_n)$$

has a unique solution modulo $m = m_1 m_2 \ldots m_n$, i.e., there is a unique solution x with $0 \leq x < m$. Furthermore, all solutions are congruent modulo m.

We can construct a solution as follows.

1. Let $m = m_1 m_2 \ldots m_n$.
2. Let $M_k = \frac{m}{m_k}$ for all $k = 1,2, \ldots \ldots n$.
3. For all $k = 1,2, \ldots \ldots n$ find integers $y_k$ such $M_k y_k \equiv 1\ (mod m_k)$

Since $\gcd(M_k, m_k) = 1$, we know that $y_k$ exists. Euclid's extended algorithm can be used to Find yk.

The integer $a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots + a_n M_n y_n$ is a solution of the system. The integer

$$x = (a_1 M_1 y_1 + a_2 M_2 y_2 + \cdots + a_n M_n y_n)\ mod m \qquad (1)$$

is the unique solution with $0 \leq x < m$.

The Mixed Radix Conversion: The residue to binary converter can be implemented using the MRC as follow

$$X = V_n \prod_{i=1}^{n} P_i + \cdots + V_3 P_2 P_1 + V_2 P_1 + V_1 \qquad (2)$$

The coefficients $V_i P$ can be obtained from residues by:

$$V_1 = x_1 \qquad (3)$$

$$V_2 = \left| (x_2 - x_1) \left| P_1^{-1} \right|_{P_2} \right|_{P_2} \qquad (4)$$

$$V_3 = \left| \left( (x_3 - x_1) \left| P_1^{-1} \right|_{P_3} - V_2 \right) \left| P_2^{-1} \right|_{P_3} \right|_{P_3} \qquad (5)$$

In general case we have

$$V_n = \left| \left( \left( (x_n - V_1) \left| P_1^{-1} \right|_{P_n} - V_2 \right) \left| P_2^{-1} \right|_{P_n} - \ldots - V_{n-1} \right) \left| P_{n-1}^{-1} \right| P_n \right|_{P_n} \qquad (6)$$

New Chinese Remainder Theorem 1: by New CRT-I with the 3-moduli set $\{P_1, P_2, P_3\}$ the number X can be computed from its corresponding residues $(x_1, x_2, x_3)$ using the following equations

$$Z = x_1 + m_1 \left| K_1 (X_2 - X_1) + K_2 M_2 (X_3 - X_2) \right|_{m_2 m_3} \qquad (7)$$

Where

$$|k_1 m_1|_{m_2 m_3} = 1 \qquad (8)$$

$$|k_2 m_1 m_2|_{m_3} = 1 \qquad (9)$$

Where $k_1, k_2$ are multiplicative inverses.

## 3. Design of Reverse Converter

For the design of reverse converter I use the new CRT theorem, the following lemmas and properties are needed for the derivation of the conversion algorithm

Theorem 1: modules are pairwise relatively prime.

Proof:

Using the Euclid's algorithm to find the greatest common divisor:

Euclid (a,b)
if b=0
return a
else
return Euclid (b,a mod b)
if a was 1 then we conclude that numbers are prime.
begin with

$\gcd(2^{2n+2} - 1, 2^{2n+1} - 1) = \gcd(2^{2n+2} - 1, 1) = 1$
so $2^{2n+2} - 1, 2^{2n+1} - 1$ are coprime.

For the moduli $\{2^{2n+1} - 1, 2^n\}$ and $\{2^{2n+2} - 1, 2^n\}$, either you can use the previous steps or noting that $2^{2n+2} - 1, 2^{2n+1} - 1$ are both odd numbers while $2^n$ is even so it is clear that $2^n$ is relatively prime with both $2^{2n+2} - 1$ and $2^{2n+1} - 1$

Lemma 1:

The multiplicative inverse of $|m_1|_{m_2 m_3}$ is

$$K_1 = \left| (2^{2n+2} - 1)^{-1} \right|_{(2^{2n+1} - 1) 2^n} \qquad (10)$$

$$K_1 = 2^{3n+1} + 2^{2n+2} - 2^n - 1 \quad (11)$$

Proof:

$$|(2^{3n+1} + 2^{2n+2} - 2^n - 1)(2^{2n+2} - 1)|_{2^{2n+1}-1,2^n} = 1 \quad (12$$

Lemma 2:

The multiplicative inverse of $|m_1 m_2|_{m_3}$ is

$$K_2 = |(2^{2n+2} - 1)(2^{2n+1} - 1)^{-1}|_{(2^{2n+1}-1)2^n} \quad (13)$$

$K_2 = 1$

Proof:

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

$$|T_1 + T_2 + T_3|_{2^n}, \quad x_2 \geq x_1 \quad (14a)$$

$$|T_1 + T_2 + +T_{31}|_{2^n}, \quad x_2 < x_1 \quad (14b)$$

Definition 1:

If a mod n $\equiv$ b then a $= c \times n + b$ When you reduce a number a modulo n you usually want $0 \leq b < n$

Definition 2:

The residue of a negative residue number (–v) in modulo $(2^n)$ is the 2's complement of v
$|-x|_{2^n} = 2'scomplement\ of\ x$

Definition 3:

The multiplication of a residue number v by $2^p$ in modulo $2^n$ is carried out by p bit left shift and zero filling right p bits, where p is a natural number.

Assuming a and b to be integers, we have the following properties

$$|aP_1|_{P_1P_2} = |a|_{P_2} * P_1 \quad (15)$$

$$\left||a|_{P_1P_2}\right|_{P_1} = |a|_{P_1} \quad (16)$$

$$|a+b|_{P_1} = \left||a|_{P_1} + |b|_{P_1}\right|_{P_1} \quad (17)$$

## 4. Conversion theorem

In this section I propose a theorem to convert the residue number $(x_1, x_2, x_3)$ into binary representation for the moduli set $\{2^{2n+2} - 1, 2^{2n+1} - 1, 2^n\}$ using the new CRT I:

$$X = x_1 + m_1 |K_1(x_2 - x_1) + K_2 m_2 (x_3 - x_2)|_{m_2 m_3} \quad (18)$$

$$X = x_1 + (2^{2n+2} - 1)|(2^{3n+1} + 2^{2n+2} - 2^n - 1)(x_2 - x_1) + (2^{2n+1} - 1)(x_3 - x_2)|_{m_2 m_3} \quad (19)$$

Where m2= $(2^{2n+1} - 1)$
And m3= $2^n$

The binary number $X = (x_1, x_2, x_3)$ is given by

$$X = x_1 + (2^{2n+2} - 1)Z \quad (20)$$

Where

$$Z = (2^{2n+1} - 1)Y + |x_2 - x_1|_{(2^{2n+1}-1)} \quad (21)$$

$$Y = |x_3 + x_2(2^n + 1) - 2x_1|_{2^n}$$

Where

$$T_1 = (x_{3,n-1} x_{3,n-2} \ldots\ldots\ldots\ldots\ldots x_{3,1} x_{3,0}) \quad (22)$$

$$T_2 = (x_{2,n-1} x_{2,n-2} \ldots\ldots\ldots\ldots\ldots x_{2,1} x_{2,0}) \quad (23)$$

$$T_{31} = (\overline{x_{1,n-2}} \ldots\ldots\ldots\ldots \overline{x_{1,1} x_{1,0}} 1) \quad (24)$$

$$T_3 = T_{31} + 1 \quad (25)$$

Proof:
The binary vectors $x_1, x_2$ and $x_3$ can be represented in bit –level as

$$x_1 = \underbrace{(x_{1,2n+1} x_{1,2n} \ldots\ldots\ldots\ldots\ldots x_{1,1} x_{1,0})}$$

$$x_2 = (\overbrace{x_{2,2n}\underbrace{x_{2,2n-1}\ldots\ldots\ldots\ldots\ldots x_{2,1}x_{2,0}}_{2n+1\text{ bits}}}^{2n+2\text{ bits}})$$

$$x_3 = (\underbrace{x_{3,n-1}x_{3,n-2}\ldots\ldots\ldots\ldots\ldots x_{3,1}x_{3,0}}_{n\text{ bits}})$$

Using the new CRT I:

$$X = x_1 + (2^{2n+2} - 1)|(2^{3n+1} + 2^{2n+2} - 2^n - 1)(x_2 - x_1) + (2^{2n+1} - 1)(x_3 - x_2)|_{m_2m_3} \quad (26)$$

$$X = x_1 + (2^{2n+2} - 1)\left|\left((2^{n+1} - 1)(2^n + 2) + 1\right)(x_2 - x_1) + (2^{2n+1} - 1)(x_3 - x_2)\right|_{m_2m_3}$$

$$X = x_1 + (2^{2n+2} - 1)|(x_2 - x_1) + (2^{2n+1} - 1)[(2^n + 2)(x_2 - x_1) + (x_3 - x_2)]|_{m_2m_3}$$

$$X = x_1 + (2^{2n+2} - 1)|(x_2 - x_1) + (2^{2n+1} - 1)[2^n x_2 + 2x_2 - 2^n x_1 - 2x_1 + x_3 - x_2]|_{m_2m_3}$$

$$X = x_1 + (2^{2n+2} - 1)|(x_2 - x_1) + (2^{2n+1} - 1)[x_2(2^n + 1) - x_1(2^n + 2) + x_3]|_{m_2m_3}$$

$$X = x_1 + (2^{2n+2} - 1)|(x_2 - x_1) + (2^{2n+1} - 1)[x_2(2^n + 1) - x_1(2^n + 2) + x_3]|_{(2^{2n+1}-1)(2^n)}$$

$$-x_1 = |2^n - x_1|_{2^n} \quad (27)$$

$$X = x_1 + (2^{2n+2} - 1)|(x_2 - x_1) + (2^{2n+1} - 1)[x_2(2^n + 1) + x_1(2^n - 2^n - 2) + x_3]|_{(21^{2n+1}-1)(2^n)}$$

$$X = x_1 + (2^{2n+2} - 1)|(x_2 - x_1) + (2^{2n+1} - 1)[x_2(2^n + 1) + x_1(-2) + x_3]|_{(2^{2n+1}-1)(2^n)}$$

$$X = x_1 + (2^{2n+2} - 1)\ Z$$

$$Z = |(x_2 - x_1) + (2^{2n+1} - 1)[x_2(2^n + 1) - 2x_1 + x_3]|_{(2^{2n+1}-1)(2^n)}$$

When $x_2 \geq x_1$

$$x_2 - x_1 = |x_2 - x_1|_{(2^{2n+1}-1)} \quad (28)$$

$$Z = (2^{2n+1} - 1)|x_3 + x_2(2^n + 1) - 2x_1|_{2^n} + |x_2 - x_1|_{(2^{2n+1}-1)}$$

$$Z = (2^{2n+1} - 1)Y + |x_2 - x_1|_{(2^{2n+1}-1)}$$

$$Y = |x_3 + x_2(2^n + 1) - 2x_1|_{2^n}$$

$$T_1 = |x_3|_{2^n} = (x_{3,n-1}x_{3,n-2}\ldots\ldots\ldots\ldots\ldots x_{3,1}x_{3,0})$$

$$T_2 = |x_2(2^n + 1)|_{2^n} = |\langle x_2\rangle\langle x_2\rangle|_{2^n}$$

$$T_2 = (x_{2,n-1}x_{2,n-2}\ldots\ldots\ldots\ldots\ldots x_{2,1}x_{2,0})$$

$$T_3 = |-2x_1|_{2^n}$$

$$T_3 = \left|-2(x_{1,n-1}\ldots\ldots\ldots\ldots x_{1,1}x_{1,0})\right|_{2^n}$$

$$|2x_1|_{2^n} = (x_{1,n-2}\ldots\ldots\ldots\ldots x_{1,1}x_{1,0}0)$$

$$T_{31} = (\overline{x_{1,n-2}}\ldots\ldots\ldots\ldots\overline{x_{1,1}x_{1,0}}1)$$

$$T_3 = T_{31} + 1$$

$$Y = |T_1 + T_2 + +T_3|_{2^n}$$

When $x_2 < x_1$

$$x_2 - x_1 = |x_2 - x_1|_{2^{2n+1}-1} - 2^{2n+1} - 1$$

$$Z = |(2^{2n+1} - 1)(x_3 + x_2(2^n + 1) - 2x_1 - 1) + |x_2 - x_1|_{2^{2n+1}-1}|_{(2^{2n+1}-1)(2^n)}$$

$$Z = (2^{2n+1} - 1)|(x_3 + x_2(2^n + 1) - 2x_1 - 1)|_{2^n} + |x_2 - x_1|_{(2^{2n+1}-1)}$$

$$Z = (2^{2n+1} - 1)Y + |x_2 - x_1|_{(2^{2n+1}-1)}$$

$$Y = |x_3 + x_2(2^n + 1) - 2x_1 - 1|_{2^n}$$

$$T_1 = |x_3|_{2^n} = (x_{3,n-1}x_{3,n-2}\ldots\ldots\ldots\ldots\ldots x_{3,1}x_{3,0})$$

$$T_2 = (x_{2,n-1}x_{2,n-2}\ldots\ldots\ldots\ldots\ldots x_{2,1}x_{2,0})$$

$$T_3 = |-2x_1 - 1|_{2^n}$$

$\mathrm{T}_3 = \mathrm{T}_{31} = (\overline{\mathrm{x}_{1,n-2}} \dots \dots \dots \dots .. \overline{\mathrm{x}_{1,1}\mathrm{x}_{1,0}}1)$

$\mathrm{Y} = |\mathrm{T}_1 + \mathrm{T}_2 + +\mathrm{T}_{31}|_{2^n}$

## 5. Example

Given n=4, then
$m_1 = 1023$
$m_2 = 511$
$m_3 = 16$
And suppose that
$x_1 = 2 = \langle 00\rangle\langle 0000\rangle\langle 0010\rangle$
$x_2 = 3 = \langle 0\rangle\langle 0000\rangle\langle 0011\rangle$
$x_3 = 1 = \langle 0001\rangle$
Then
$T_1 = |X_3|_{2^n} = \langle 0001\rangle$
$T_2 = (x_{2,n-1}x_{2,n-2} \dots \dots \dots \dots \dots .. x_{2,1}x_{2,0}) = \langle 0011\rangle$
And noting that $x_2 \geq x_1$
$T_{31} = (\overline{x_{1,n-2}} \dots \dots \dots \dots .. \overline{x_{1,1}x_{1,0}}1) = \langle 1011\rangle$
$T_3 = T_{31} + 1 = \langle 1100\rangle$
$Y = |1 + 3 + +12|_{2^n} = \langle 0000\rangle$
$Z = (2^{2n+1} - 1)Y + |X_2 - X_1|_{(2^{2n+1}-1)}$
$Z = \underbrace{\langle 0000\rangle\langle\ 0\ \ 0000\ \ 0001\rangle}_{\text{3n+1 bits}}$

$X = X_1 + (2^{2n+2} - 1)\, Z$

$X_1 + 2^{2n+2}\, Z$
$= \langle 0000\rangle\langle\ 0\ \ 0000\ \ 0001\rangle\langle 00\rangle\langle 0000\rangle\langle 0010\rangle$
$+$
$-Z\{\, Z\ 2's\ complement\ \}$
$= \langle 1111\rangle\langle\ 1\ \ 1111\ 1111\rangle\langle 11\rangle\langle 1111\rangle\langle 1111\rangle$

$X = \langle 0000\rangle\langle\ 0\ \ 0000\ \ 0001\rangle\langle 00\rangle\langle 0000\rangle\langle 0001\rangle$
$= 1025$

Another example if we take

$x_1 = 931 = \langle 11\rangle\langle 1010\rangle\langle 0011\rangle$
$x_2 = 423 = \langle 1\rangle\langle 1010\rangle\langle 0111\rangle$
$x_3 = 0 = \langle 0000\rangle$
Noting that $x_2 < x_1$
$T_1 = |X_3|_{2^n} = \langle 0000\rangle$
$T_2 = (x_{2,n-1}x_{2,n-2} \dots \dots \dots \dots \dots .. x_{2,1}x_{2,0}) = \langle 0111\rangle$
$T_{31} = (\overline{x_{1,n-2}} \dots \dots \dots \dots .. \overline{x_{1,1}x_{1,0}}1) = \langle 1001\rangle$
$Y = |7 + 9 + +0|_{2^n} = \langle 0000\rangle$
$Z = (2^{2n+1} - 1)Y + |X_2 - X_1|_{(2^{2n+1}-1)}$
$Z = \langle 0000\rangle\langle\ 0\ \ 0000\ \ 0011\rangle$
$X = X_1 + (2^{2n+2} - 1)\, Z$

$= \langle 0000\rangle\langle\ 0\ \ 0000\ \ 0011\rangle\langle 11\rangle\langle 1010\rangle\langle 0011\rangle +$
$\langle 1111\rangle\langle\ 1\ \ 1111\ 1111\rangle\langle 11\rangle\langle 1111\rangle\langle 1101\rangle +$
$= \langle 0000\rangle\langle\ 0\ \ 0000\ \ 0011\rangle\langle 11\rangle\langle 1010\rangle\langle 0000\rangle$
=4000

## 6. Hardware Implementation

The proposed reverse converter is based on Eq. (13), Eq. (14), Eq.(15 a) and Eq.(15 b). The hardware implementation of Eq.(15) requires n-bits carry save adder to reduce these three input numbers into two numbers namely, sum and carry then n-bits module adder is used to add the sum and carry together to generate Y.
Let us denote the module that computes Y as Module Y, which employ one n-bits CSA the first used to compute then n-bits carry propagate adder (CPA) with end around carry (EAC) act as n-bits module adders, as shown in Fig. 1
Figure 2 show the execution of Eq. (14) we need a subtractor for $\mathrm{x}_2 - \mathrm{x}_1$ which can be implemented by (2n+1) bits CSA with EAC, this EAC bit can indicate whether $\mathrm{x}_2 \geq \mathrm{x}_1$ or not so can be used as control bit for n-bits 2-to-1 multiplexer that decide the correct Y ;
$\mathrm{Y} = |\mathrm{T}_1 + \mathrm{T}_2 + +\mathrm{T}_3|_{2^n}$ when $\mathrm{x}_2 \geq \mathrm{x}_1$ and $\mathrm{Y} = |\mathrm{T}_1 + \mathrm{T}_2 + +\mathrm{T}_{31}|_{2^n}$ when $\mathrm{x}_2 < \mathrm{x}_1$.

To complete our implementation of Eq. (14) we simply concatenate Y with SUB1 output into a (3n+1) bits number then another subtractor SUB2 is used to generate Z

Finally for the implementation of Eq. (13) a (5n+3) bits number is generated by concatenation of Z and $\mathrm{X}_1$ and using SUB3 to subtracts Z we can get the final X as shown in Fig. 2.

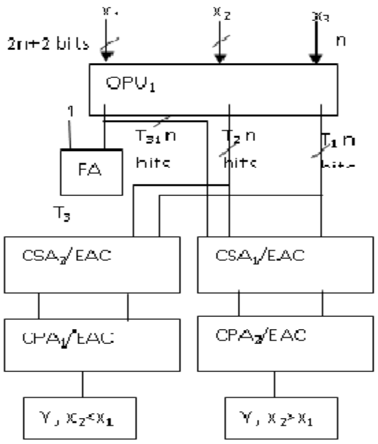


Fig. 1 Module Y: detailed architecture

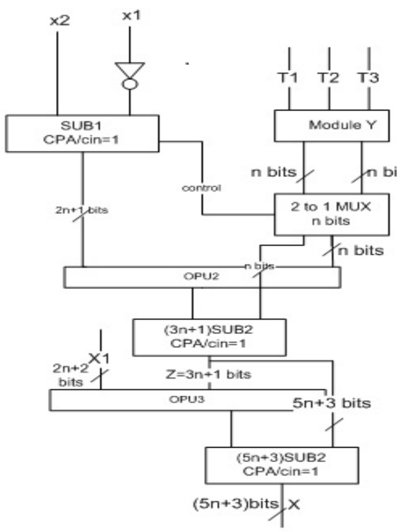


Fig. 2 Proposed Convertor

The operands preparation unit of Fig. 1 contains (n) not gates for doing the inversion of $x_1$ to find its one's complement.

It is important to know that some parts of equation 13 and 14 can be implemented simply by concatenation without the use of any calculative hardware in Fig.3 we represent this by using OPU2, OPU3

Note that $SUB_2$ is (3n+1)-bit binary subtractor, which employ (3n+1) FA's and n-bit not gates to find the inversion of Y to compute subtraction.

Also since we have (2n+1) bits of 1's, (2n+1) FA's in $SUB_2$ can be reduced to a pairs of XNOR/OR gates. Again for the implementation of $SUB_3$ a (5n+3) regular binary subtractor is used where the (3n+1) of the (5n+3) FA's can be replaced by a pairs of XNOR/OR gates.

## Performance evaluation

The primary digital charcterstics of any digtal design are the speed, area and power. The speed can be computed by throughput,latency and timing; the latecny is the time between data input and the processing data outputs while timing is the logical delay between elements, when a design doesn't meet timing it means that the delay of the critical path is larger than the target clock period.

So to optimize the performance efficiently in your design you have to reduce the delay in critical path one way to do this by considering the amount of parallism between entities and reduce the dependencies among them.
Parallism was implemented in the new CRT theorem where in the conversion process; the weighted number can be retrieved faster because the operations are done in parallel, without depending on other results.

In this section, hardware requiremnt and speed of the proposed reverse converter based on our moduli set is studied.

Firstly we must calculate the hardware reuirment and delay of the proposed converter. Then compare the result with other converters from both hardware cost and delay viwepoints.

In Table 1 the complexity and delay introduced by different adders and gates used in the proposed converter are listed.

Table 1: Complexity and Delay of Various Components

| Parts | FA | NOT | XNOR/OR pairs | Delay |
|---|---|---|---|---|
| OPU1 | | (n-1) | | $t_{not}$ |
| Adder | 1 | | | $t_{FA}$ |
| CSA1 | n | | | $t_{FA}$ |
| CSA2 | n | | | $t_{FA}$ |
| CPA1 | n | | | $nt_{FA}$ |
| CPA2 | n | | | $nt_{FA}$ |
| SUB1 | 2n+1 | 2n+2 | | $(2n+1)\ t_{FA}+\ t_{not}$ |
| MUX | n | | | $t_{FA}$ |
| SUB2 | n | n | 2n+1 | $(3n+1)\ t_{FA}+\ t_{not}$ |
| SUB3 | 3n+1 | 3n+1 | 2n+2 | $(5n+3)\ t_{FA}+\ t_{not}$ |
| Total | 11n+3 | 7n+2 | 4n+3 | $(12n+9)t_{FA}+4t_{not}$ |

To improve the throughput rate, pipelining is usually applied in real implementation, the delay of Module Y is smaller than that of $SUB_1$, and hence the delay of the converter depends on the delay of the critical path consisting of $SUB_1$, MUXs, $SUB_2$, and $SUB_3$. The delay of a CSA or a MUX is the same as that of an FA, namely, $t_{FA}$. The delay of each of the modulo adders in Module Y is $nt_{FA}$ and that of the modulo subtractor $SUB_1$ is $(2n+1)\ t_{FA}+$ $t_{not}$, while the delay of the binary subtractor $SUB_2$ is

(3n+1) $t_{FA}$+ $t_{not}$, and that of $SUB_3$ is (5n+3) $t_{FA}$+ $t_{not}$ Thus, the converter has a total delay of

$$\begin{aligned} \mathrm{Delay} = (2n+1)t_{FA} + t_{not}SUB_1 + t_{FA}MUX \\ + (3n+1)t_{FA} + t_{not}SUB_2 \\ + (5n+3)t_{FA} + t_{not}SUB_3 \end{aligned}$$

$$Total\ Delay = (10n+6)t_{FA} + 3t_{not}$$

So as described above doing the computations simultaneously in parallel improves the critical path delay ,Table II describe the improvement in delay that was achieved using the new CRT for the conversion process in comparison with convertors have the same or less dynamic range.

*Table 2:* Delay Comparison between the Proposed Reverse Convertor and Related Works

| Converter | DR(bits);n=4 | Delay($t_{FA}$) |
|---|---|---|
| [12] | 16 | 16n+22 |
| [22] | 22 | 14n+8 |
| [28] | 19 | 18n+17 |
| proposed | 22 | 10n+6 |

Hence, converters based the New CRT's require no big size modulo adders. In many cases, only one modulo operation is needed. The numbers involved in the conversion are smaller than the numbers in the CRT. This will gain speed, since binary arithmetic speed is often bounded by the size of the numbers. But, New CRTs are hardware intensive as they require many inverse modulus operators, modulus operators, multipliers and dividers. Dividers and inverse modulus operators in turn needs many half and full adders and subtractors.but since hardware cost has been driven low nowadays, illustration of the idea that there is always room for better performance.

In order to obtain comperhensive view of the improvement in implementaion we extend TableI I for different values of n as shown in Table III.since converter of [13] has 4n bits dynamic range we exclude it from comprsion.

Table 3: Performance Comparison Using Different Values of n

| N | [22] | | [28] | | Proposed | |
|---|---|---|---|---|---|---|
| | DR (bits | Delay ($t_{FA}$) | DR (bits) | Delay ($t_{FA}$) | DR (bits) | delay ($t_{FA}$) |
| 4 | 22 | 64 | 19 | 89 | 22 | 46 |
| 8 | 62 | 120 | 39 | 161 | 62 | 86 |
| 12 | 82 | 176 | 59 | 233 | 82 | 126 |
| 16 | 102 | 232 | 79 | 305 | 102 | 166 |
| 20 | 122 | 288 | 99 | 377 | 122 | 206 |

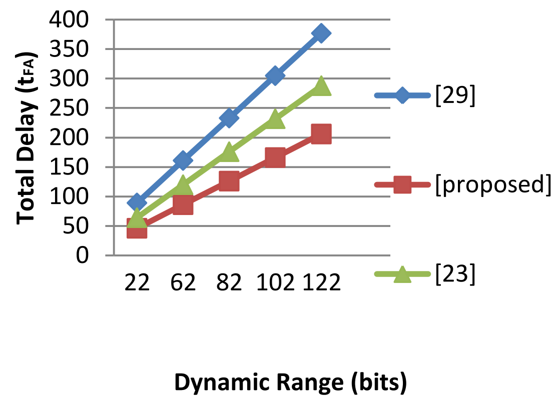


Fig. 3 comparison of delay for different converters

Based on the specific sets ***Mi*** -8, ***Mi*** -16, ***Mi*** -32 and ***Mi*** 64, the corresponding n that represent each converter found and delay of each moduli set for all converters ***Ci*** - 8,Ci-16, Ci-32 and Ci-64 is computed. It is assumed that $C_1$={$2^{2n+2}-1, 2^{2n+1}-1, 2^n$} based on MRC and $C_2$={$2^n, 2^{n-1}-1, 2^{n+1}-1, 2^n-1, 2^n+1$}, finally $C_3$is the proposed converter which is for {$2^{2n+2}-1, 2^{2n+1}-1, 2^n$} based on CRTI , for 64-bit converter n=13, for both.

Table 4: Specific seytsm_I-8,M_I-16,M_I-32,(I-1,…,4)

| Converter | 8-bit $M_i-8$ | 16-bit $M_i-16$ | 32-bit $M_i-32$ |
|---|---|---|---|
| $C_{1.1}$ | {3,4,5,7,1} | {15,16,17,31,7} | {127,128,129,255,63} |
| $C_{1.2}$ | {63,31,4} | {225,127,8} | {16383,8191,64} |

Table 5: Delay for Specific Dynamic Range

| Converter | $C_i-8$ | $C_i-16$ | $C_i-32$ | $C_i-64$ |
|---|---|---|---|---|
| $C_{1.1}$ | 53/n=2 | 89/n=4 | 143/n=7 | 251/n=13 |
| $C_{1.2}$ | 36/n=2 | 50/n=3 | 120/n=8 | 190/n=13 |
| $C_{1.3}$ | 26/n=2 | 36/n=3 | 86/n=8 | 136/n=13 |

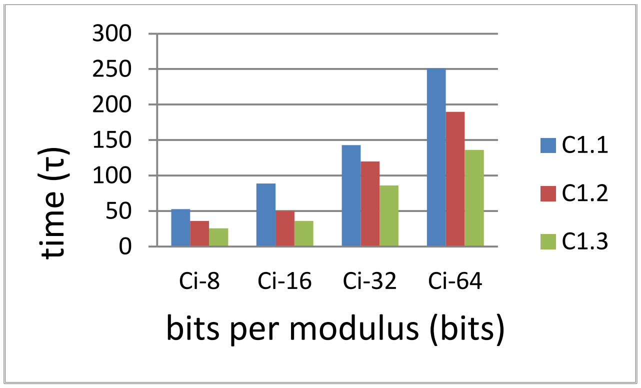


Fig. 4 Time Performance Comparison vs. dynamic range

## Conclusion

A new converter for specific moduli set $\{2^{2n+2}-1, 2^{2n+1}-1, 2^n\}$ was proposed using New Chinese Remainder Theorem I. The design is compact and provides higher speed of conversion compared to other implementations that operate on the same set. The hardware requirements for the proposed converter are comparable to similar converters. The manipulation technique presented in this paper can serve as a guideline for similar design procedure and will open up many doors for further RNS research. It is expected more efficient arithmetic algorithms can be developed based on it, and many converters as the one proposed here can be implemented.

## References

[1] [1] Bajard, Claude, and Plantard(2004) "RNS bases and conversions". proceedings of SPIE , 2004,Vol. 5559,Page:61-75.

[2] Bhardwaj, Srikanthan,and Clarke.(1999)"A reverse converter for the 4-moduli superset {2n-1, 2n, 2n+ 1, 2n+1 +1}." Computer arithmetic proceedings of 14th IEEE Symposium , 1999,Pages:168-175.

[3] Hiasat, and Sweidan(2003) "Residue number system to binary converter for the moduli set (2n− 1, 2n − 1, 2 n+ 1)". Journal of systems architecture ,2003,Vol.49,Pages: 53-58.

[4] Hiasat, and Zohdy (1998)"Residue-to-binary arithmetic converter for the moduli set (2k, 2k-1, 2 k-1-1)". Circuits and Systems II: Analog and Digital Signal Processing of IEEE Transactions, 1998, Vol. 45,Pages: 204-209.

[5] Hariri, Arash, Navi, and Rastegar(2008)"A new high dynamic range moduli set with efficient reverse converter" .Computers & mathematics with applications , 2008, Vol.55, Pages: 660-668.

[6] Jameii, Mahdi, Taghipour, and Azad(2011) "Using both Binary and Residue Representations for Achieving Fast Converters in RNS".journal of advances in computer research ,2011, Pages: 91-104.

[7] Jenkins (1978)"Techniques for residue-to-analog conversion for residue-encoded digital filters". Circuits and Systems, IEEE Transactions , 1978,Vol. 25,Pages: 555-562.

[8] Lewis, Michael, Jon Mellott, and Taylot(2004) "An efficient residue to analog converter", IEEE International Conference , 2004 ,Vol. 5 ,Pages:157-60.

[9] Modiri, Samira, Movaghar, and Barati(2012) "Study of error control capability for the new moduli set {22n+ 1+ 2n-1, 22n+ 1-1, 2n-1, 23n, 23n+ 1-1}". Journal of Advanced Computer Science & Technology ,2012, Vol. 1, Pages: 176-186.

[10] Mohan,and Ananda(2002) "Residue number systems: algorithms and architectures". Springer, 2002,vol.1,page:1-268.

[11] Mohan, and Ananda.(2007)"RNS-to-binary converter for a new three-moduli set $\{2^{n+1}-1, 2^n, 2^{n-1}\}$". Circuits and Systems II: Express Briefs, IEEE Transactions ,2007, Vol.54, Pages: 775-779.

[12] Mohan, Ananda, and Premkumar(2007) "RNS-to-binary converters for two four-moduli sets $\{2^{n-1}, 2^n, 2^{n+1}, 2^{n+1}-1\}$ and $\{2^{n-1}, 2^n, 2^{n+1}, 2^{n+1}+1\}$." Circuits and Systems I: Regular Papers, IEEE Transactions ,2007, Vol. 54, Pages: 1245-1254.

[13] Molahosseini, and Sabbagh (2011)"Efficient Residue to Binary Conversion Based on a Modified Flexible Moduli Set" . Proceedings of AIP Conference ,2011, Vol. 1389,Pages: 1066-1069.

[14] Molahosseini, Sabbagh, Dadkhah, and Navi(2009) "A new five-moduli set for efficient hardware implementation of the reverse converter" .IEICE Electronics Express ,2009, Vol.6, Pages: 1006-1012.

[15] Navi, Keivan, Molahosseini, and Esmaeildoust (2011)"How to teach residue number system to computer scientists and engineers." IEEE Transactions ,2011,Vol.54, Pages: 156-163.

[16] Omondi, Amos, and Premkuar.( 2007) "Residue number systems: theory and implementation". Imperial College Press, 2007.

[17] Pham, Premkumar, and Madhukumar(2010)"Reduced complexity analogue-to-residue conversion employing folding number system" .Circuits, Devices & Systems,2010,Vol. 4,Pages: 30-41.

[18] Premkumar,Ang, and Lai(2006) "Improved memoryless RNS forward converter based on the periodicity of residues". Circuits and Systems II: Express Briefs, IEEE Transactions,2006 ,Vol. 6, Pages: 133-137.

[19] Radhakrishnan, and Preethy (1998) "A new approach to data conversion: direct analog-to-residue converter". Acoustics, Speech and Signal Processing, IEEE International Conference,1998,Vol. 5, Pages:3013 – 3016.

[20] Radhakrishnan and Preethy (1999) "A parallel approach to direct analog-to-residue conversion".Information processing letter,1999,Vol. 69, Pages: 249-252.

[21] Riazi, Sahel, Hassanpour, and Hosseinzadeh. (2011)"An efficient architecture of residue to binary converter for new four-moduli set".International Journal ,2011,Vol. 2,Pages: 709-715.

[22] Samiria,Moraghar and Barati.(2012)"An Efficient Reverse Converter for the New Modulus Set{ $2^{2n+2}-1$, $2^{2n+1}-1$, $2^n$}". International Journal of Advanced Research in Computer Science and Software Engineering,2012,Vol.2Pages:447-452

[23] Shende, Radha, and Zode.( 2012)"Efficient design 2 k− 1 binary to residue converter." Devices, Circuits and Systems (ICDCS), International Conference IEEE, 2012, Pages: 482 - 485

[24] Taheri, MohammadReza, Pirhoseinlo,Esmaeildoust,and Navi. (2012)"High speed reverse converter for high dynamic range moduli set." International Journal of Advances in Engineering & Technology,2012,vol.3.

[25] Taheri, Reza, Pirhoseinlo,Esmaeildoust,and Navi.(2012)"Efficient reverse converter design for five moduli set {$2^n$, $2^{2n+1}$-1, $2^{n/2}$-1, $2^{n/2}$+1,$2^{n+1}$}". Journal of Computations & Modelling ,2012, Vol.2,Pages: 93-108.

[26] Wang.(2003) " study of the residue-to-binary converters for the three-moduli sets" .Circuits and Systems I: Fundamental Theory and Applications, IEEE, 2003,Vol. 50, Pages: 235-243.

[27] Wang,Swamy, and Wang . (2000)"A high-speed residue-to-binary converter for three-moduli (2k, 2k-1, 2 k-1-1) RNS and a scheme for its VLSI implementation". Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions , 2000,Vol.47,Pages: 1576-1581.