I.J. Image, Graphics and Signal Processing, **2012, 9, 37-43**
Published Online September 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2012.09.06

Effective Reverse Converter for General Three Moduli Set$\{2^n-1,2^n+1,2^{pn+1}-1\}$

Mehdi Hosseinzadeh[a], Keihaneh Kia
Department of Computer Engineering, Science and Research Branch, Islamic Azad University
Tehran, Iran
[a]hosseinzadeh@sr.iau.ac.ir

Abstract— Residue number system is a non-weighted integer number system which uses the residues of division of ordinary numbers by some modules for representing that ordinary numbers. In this paper, the general three moduli set $\{2^n-1,2^n+1,2^{pn+1}-1\}$ based on CRT algorithm is proposed in which "p" is an even number greater than zero. The special case of this set for p=2 which is $\{2^n-1,2^n+1,2^{2n+1}-1\}$ is also described in this paper. Since the dynamic range of this set is odd, some difficult problems in RNS can be easily solved based on this set using parity checking. The proposed reverse converter is better in speed and hardware in comparison to reverse converters in similar dynamic range. Moreover, from the complexity point of view, the internal arithmetic circuits of this moduli set is improved and is less complex than the other sets in similar dynamic range.

Index Terms— Reverse Converter, Moduli Set, Dynamic Range, Residue Number System

I. Introduction

Residue number system is a non-weighted number system which is specified with a muduli set $\{m_1,m_2,\ldots,m_n\}$ in which an integer number X is represented as $(x_1,x_2,\ldots,x_n)$ that $x_i=x \bmod m_i$. Arithmetic operations on residues can be performed in each moduli in parallel without carry propagation between them. The RNS has been widely considered for efficient hardware implementation of digital signal processing (DSP) [3], and for the implementation of high-speed FIR filters [4]. Moreover, RNS has applications in image processing systems, especially RNS image coding which can offer high-speed VLSI implementation of secure image processing algorithms [5].

Because of the carry free property of some operations like addition, subtraction and multiplication, implementation of these operations are easy and fast. Some operations like sign determination, number comparison, and overflow detection cannot be accomplish free of carry between muduli, so they are considered as fundamental problems in RNS. Different solutions are proposed to accomplish these operations. One of them is parity checking where parity means the residue in redundant modulo 2. But this solution is easily to implement only when the dynamic range is odd.

Moduli set selection and reverse conversion design is very significant in RNS. Reverse conversion is mainly implemented with one of the algorithms of Chinese reminder theorem and mixed-radix conversion or a combination of these two.

One of the most popular moduli sets is $\{2^n-1,2^n,2^n+1\}$. But today its dynamic range is not sufficient for many applications. So moduli sets with larger dynamic range and sets with more moduli for increasing parallelism are mostly proposed. The modulo 2^n looks an appropriate modulo with respect to hardware cost and delay of arithmetic circuits and converters. But this modulo is even therefore the dynamic range will be even, so we would not be able to use parity checking as a solution for the fundamental problems in RNS.

Few moduli sets with odd dynamic ranges were also proposed, like: $\{2^n-1,2^n-3,\ 2^n+3,\ 2^n+1\}$[6], $\{2^n-1,2^n+1,2^{2n}-2,2^{2n+1}-3\}$[7], $\{2^n-1,2^n+1,2^{2n}+1\}$ [8], $\{2^{n/2}-1,2^{n/2}+1,2^n+1,2^{2n+1}-1\}$[16]. Because of using 2^n-2 and 2^n+3 moduli, complexity of internal arithmetic circuits for [6], [7] is high. In comparison to our reverse converter, since in moduli set $\{2^n-1,\ 2^n+1,\ 2^{2n}+1\}$ the third modulo is a multiple of the other moduli, the reverse converter has a better performance. But the arithmetic circuits for moduli in the form of 2^n+1 are complex and unfortunately two of them are in this moduli set, so the performance has decreased in the overall RNS. In moduli set $\{2^{n/2}-1,2^{n/2}+1,2^n+1,2^{2n+1}-1\}$, the parallelism is increased but there are unbalance moduli and also two moduli in the form of 2^n+1 which lead to decrease in the performance of overall RNS.

In this paper the reverse converter for the general three odd moduli set is proposed based on CRT algorithm in which p is an even number greater than zero. Taking p as a variant we can have the appropriate dynamic range. Therefore, the dynamic range is odd, and consequently the proposed set is amenable to solve difficult RNS problems using parity checking.

In the rest of the paper we will see a brief introduction of RNS (Section II), design of proposed converter for the general case and then for the special case (section III) and at the end of the paper we will review the performance evaluation and finally the conclusion.

II. BACKGROUND

A residue number system is defined in terms of relatively prime moduli set $\{P_1,P_2, \ldots,P_n\}$ that is gcd $(P_i,P_j) = 1$ for $i \neq j$, where gcd (P_i,P_j) denotes the greatest common divisor of P_i and P_j.

Weighted number X can be represented as $X=(x_1,x_2, \ldots ,x_n)$, where $x_i = X \bmod P_i = |X|_{P_i}$, $0 \le x_i < P_i$.

Such a representation is unique for any integer X in the range [0,M-1], where $M=P_1P_2 \ldots P_n$ is the dynamic range of the moduli set $\{P_1,P_2, \ldots,P_n\}$.

The residue number system (RNS) is a carry-free number system which can support parallel and high-speed arithmetic. In this system, a weighted number is converted into a set of small Residues and arithmetic operations can be performed in parallel on each modulo. Since arithmetic operations can be performed without carry propagation between residues, RNS leads to high-speed addition, subtraction and multiplication.

In order to perform some arithmetic operations on a weighted number In an RNS system, a converter is needed to decompose a weighted binary number into a residue represented number, with regard to the moduli set. That converter is a binary to residue converter (forward converter). After forward conversion, arithmetic operations can be performed on each modulo independently and simultaneously and without carry propagation between residues. In order to use the result of arithmetic operations in the form of a weighted number, the resulted RNS number must be converted into its equivalent weighted binary number by residue to binary conversion (reverse conversion).

Binary to residue conversion can be implemented with multi-operand modular adders simply. The arithmetic unit includes modular arithmetic circuits for each modulo channel. Reverse conversion involves a significant degree of complexity.

The algorithms of residue to binary conversion are mainly based on chinese remainder theorem (CRT) and mixed-radix conversion (MRC).

In CRT [1], the residue number $(x_1,x_2,\ldots,x_n)$ with moduli set $\{m_1,m_2,\ldots,m_n\}$ is obtained as follow:

$$X = \left| \sum^{L} \left| X_i N_i \right|_{m_i} M_i \right|_M \tag{1}$$

$$M = \prod_{i=1}^{L} m_i \tag{2}$$

Where $M_i=M/m_i$ and $N_i = |M_i^{-1}|_{Pi}$ is the multiplicative inverse of M_i modulo mi for $i=1,2,\ldots,l$.

By MRC algorithm [1], the residue represented number $(x_1,x_2,\ldots,x_n)$ can be converted into the weighted number X with moduli set $\{m_1,m_2,\ldots,m_n\}$ as follow:

$$X = a_n \prod_{i=1}^{L} m_i + \ldots + a_2 m_1 + a_1 \tag{3}$$

The cofficients a_is can be obtained from the residues by

$$a_n = \left| (((x_n - a_1)\left| m_1^{-1} \right|_{m_n} - a_2)\left| m_2^{-1} \right|_{m_n} - \ldots - a_{n-1})\left| m_{n-1}^{-1} \right|_{m_n} \right|_{m_n} \tag{4}$$

Where n>1 and $a_1=x_1$.

The RNS has many applications in digital signal processing (DSP) , image processing, RSA algorithm and communication systems. Also, RNS offers new approaches to the design of the error detection and error correction codes. The basic arithmetic components in arithmetic logic unit (ALU) and DSP systems, such as number comparison, parity checking, base extension, sign determination, and overflow detection, turn to a tough obstacle in RNS, which limit many RNS based applications.

Some difficult problems in RNS like number comparison, sign determination, and overflow detection, can be solved based on parity checking. Moreover, parity checking is also one of the fundamental issues for the division and scaling in RNS. For the odd moduli set, the parity checking is one of the fundamental issues [9].

Each RNS system is based on a moduli set which consist of a set of relatively prime integers. The majority of the algorithms for performing these difficult operations are based on reverse conversion. Hence, an efficient design of reverse converter greatly simplify the hardware implementation of these difficult operations.

The complexity of the residue to binary converter and also the speed of the RNS arithmetic circuits are mainly based on the form and the quantity of the moduli in a moduli set.

The most used moduli set is $\{2^n-1, 2^n, 2^n+1\}$ [10]. The implementation of reverse conversion, modular addition, and multiplication of this moduli set are not complex generally, but number comparison, sign determination, and overflow detection cannot be accomplished based on parity checking because the dynamic range is even. In this case, we would not be able to use parity checking as a solution for the fundamental problems in RNS.

III. DESIGN OF REVERSE CONVERTER

A. Design of Reverse Converter for General Three Moduli Set {2^n-1,2^n+1,2^{pn+1}-1}

For design of the reverse converter we use CRT algorithm. Theorems, properties and lemmas are used in the design.

Theorem1: 2^n-1, 2^n+1, 2^{pn+1}-1 are pair-wise relatively prime numbers.

Proof: based on Euclid's Theorem gcd(a,b) = gcd(b,a mod b) in which gcd(a,b) represent greater common devisor between a ,b. if a and b are relatively prime to each other, their greater common divisor is equal to one.

gcd(2pn+1-1,2n-1)=gcd(2n-1,1)=1 (5)

gcd(2pn+1-1,2n+1)=gcd(2n+1,1)=1 (6)

as 2^n-1, 2^n+1 are prime relative to 2^{pn+1}-1 and gcd(2^n+1, 2^n-1)=1, so we can conclude that all three moduli are relatively prime to each other.

Lemma 1: the multiplicative inverse of (2^n+1)×(2^{pn+1}-1) modulo 2^n-1 is 2^n-1.

Proof: since we have

$$\left|2^{pn+1}-1\right|_{2^n-1}=\left|2?\ \overbrace{2^n-1+1)?\ ..?\ 2^n-1+1)}^{p}-1\right|_{2^n-1}=\left|2-1\right|_{2^n-1}=1 \quad (7)$$

With replacing above result in equation (8), we have:

$$|M_1^{-1}\times(2^n+1)\times(2^{pn+1}-1)|_{2^n-1}=|2^{n-1}\times 2^1\times 1|_{2^n-1}=|2^n|_{2^n-1}=1 \quad (8)$$

Lemma 2: the multiplicative inverse of (2^n-1)×(2^{pn+1}-1) modulo 2^n+1 is 2^n-1.

Proof: since p is an even number, we have:

$$\left|2^{pn+1}-1\right|_{2^n+1}=\left|2?\ \overbrace{2^n+1-1)?\ ..?\ 2^n+1-1)}^{p}-1\right|_{2^n+1}=\left|2-1\right|_{2^n+1}=1 \quad (9)$$

Replacing above result in equation (10), we have:.

$$\left|M_2^{-1}?\ 2^n-1)?\ 2^{pn+1}-1)\right|_{2^n+1}=\left|2^{n-1}?\ -2^1)?\ \right|_{2^n+1}=\left|-2^n\right|_{2^n+1}=1 \quad (10)$$

Lemma 3: the multiplicative inverse of (2^n-1)×(2^n+1) modulo 2^{Pn+1}-1 is $-(2^{(p-2)n+1}+2^{(p-4)n+1}+\ldots+2^{2n+1}+2)$.

By substituting above result in equation (11), we have:

$$\left|M_3^{-1}?\ 2^n-1)?\ 2^n+1)\right|_{2^{pn+1}-1}=\left|(-2^{(p-2)n+1}-2^{(p-4)n+1}-\ldots-2^{2n+1}-2)?\ 2^{2n}-1)\right|_{2^{pn+1}-1}=$$
$$\left|(-2^{pn+1}-2^{(p-2)n+1}-\ldots-2^{2n+1})+(2^{(p-2)n+1}+2^{(p-4)n+1}+\ldots+2^{2n+1}+2)\right|_{2^{pn+1}-1}=$$
$$\left|-(2^{pn+1}-1)+1\right|_{2^{pn+1}-1}=1 \quad (11)$$

Property 1: the multiplication of residue number v by 2^p in modulo 2^n-1 is equivalent to p bit circular left shifting, where p is a natural number. The proof is mentioned in [1].

Property 2: the negative residue number (-v) in modulo 2^n-1, is equal to the one's complement of v where $0 \le v < 2^n-1$. The proof is mentioned in [1].

Based on CRT algorithm, the residue number ($x_1,x_2,\ldots,x_n$) with moduli set {$m_1,m_2,\ldots,m_n$} , is obtained as follow:

$$X=\left|\sum_{i=1}^{L}\left|X_iN_i\right|_{m_i}M_i\right|_M \quad (12)$$

$$M=\prod_{i=1}^{L}m_i \quad (13)$$

$M_i=M/m_i$ and where $N_i=|M_i^{-1}|_{pi}$ is the multiplicative inverse of M_i modulo m_i for $i=1,2,\ldots,l$.

So for the moduli set we have

$$X=|x_1\times m_2\times m_3\times M_1^{-1}+x_2\times m_1\times m_3\times M_2^{-1}+x_3\times m_1\times m_2\times M_3^{-1}|_M \quad (14)$$

Since

$$m_1\times m_2\times M_3^{-1}=(2^{2n}-1)\times(-2^{(p-2)n+1}-2^{(p-4)n+1}-\ldots-2^{2n+1}-2)$$
$$=(-2^{pn+1}-2^{(p-2)n+1}-\ldots-2^{2n+1})+(2^{(p-2)n+1}+2^{(p-4)n+1}+\ldots+2^{2n+1}+2)=$$
$$-(2^{pn+1}-1)+1=-m_3+1 \quad (15)$$

By substituting the above result in (14), we have:

$$X=|x_1\times m_2\times m_3\times M_1^{-1}+x_2\times m_1\times m_3\times M_2^{-1}+x_3\times(-m_3+1)|_M \quad (16)$$

According to [1], we can consider X as follow:

$$X=\left[\frac{X}{m_3}\right]\times m_3+x_3 \quad (17)$$

$$\left[\frac{X}{m_3}\right]=|x_1\times m_2\times M_1^{-1}+x_2\times m_1\times M_2^{-1}-x_3|_{m_1\times m_2} \quad (18)$$

$$\left\lfloor\frac{X}{m_3}\right\rfloor=\left|x_1?\ 2^n+1)?\ ^{n-1}+x_2?\ 2^n-1)?\ ^{n-1}-x_3\right|_{2^{2n}-1}=$$

$$\left|2^{n-1}(x_1x_1)+2^{n-1}(2^nx_2-x_2)-x_3\right|_{2^{2n}-1}=\left|v_1+v_2+v_3+v_4\right|_{2^{2n}-1} \quad (19)$$

With respect to moduli set{2^n-1,2^n+1,2^{pn+1}-1}, the residues (x_1,x_2,x_3) has representations in binary form as follow:

$$x_1=(x_{1,n-1}x_{1,n-2}\ldots x_{1,0}) \quad (20)$$

$$x_2=(x_{2,n}x_{2,n-1}\ldots x_{2,0}) \quad (21)$$

$$x_3=(x_{3,pn}x_{3,pn-1}\ldots x_{3,0}) \quad (22)$$

We can write equation (17) in the form of $X=Y\times m_3+x_3$, in which $Y=|v_1+v_2+v_3+v_4|_{2^{2n}-1}$.

$$v_1=\left|2^{n-1}(x_1x_1)\right|_{2^{2n}-1}=(x_{1,0}x_{1,n-1}x_{1,n-2}\ldots x_{1,0}x_{1,n-1}x_{1,n-2}\ldots x_{1,1}) \quad (23)$$

$$v_2=\left|2^{2n-1}(0\ldots0\,x_2)\right|_{2^{2n}-1}=(x_{2,0}\underbrace{0\ldots0}_{n-1}x_{2,n}x_{2,n-1}\ldots x_{2,1}) \quad (24)$$

$$v_3=\left|-2^{n-1}(0\ldots0\,x_2)\right|_{2^{2n}-1}=(\overline{x_{2,n}}\,\overline{x_{2,n-1}}\ldots\overline{x_{2,0}}\underbrace{1\ldots1}_{n-1}) \quad (25)$$

$$v_4=\left|-(x_{3,pn}x_{3,pn-1}\ldots x_{3,0})\right|_{2^{2n}-1}=$$
$$\left|-2^{pn}x_{3,pn}-2^{(p-2)n}(x_{3,pn-1}\ldots x_{3,(p-2)n})-\ldots-2^{2n}(x_{3,4n-1}\ldots x_{3,2n})-(x_{3,2n-1}\ldots x_{3,0})\right|_{2^{2n}-1} \quad (26)$$

$$v_4 = v_{4,1} + v_{4,2} + \ldots + v_{4.(p/2)+1} \quad (27)$$

$$v_{4,1} = (\underbrace{1\ldots1}_{2n-1}\overline{x_{3,pn}}) \quad (28)$$

$$v_{4,2} = (\overline{x_{3,pn-1}}\,\overline{x_{3,pn-2}}\ldots\overline{x_{3,(p-2)n}}) \quad (29)$$

$$v_{4.(p/2)+1} = (\overline{x_{3,2n-1}}\,\overline{x_{3,2n-2}}\ldots\overline{x_{3,0}}) \quad (30)$$

Since $\left|(\underbrace{1\ldots1}_{2n})\right|_{2^{2n}-1} = \left|(2^{2n}-1)\right|_{2^{2n}-1} = 0$,we can simplify v_3 and $v_{4,1}$ to one vector that is introduce as v' in equation (31).

$$v' = v_3 + v_{4,1} = (\overline{x_{2,n}}\,\overline{x_{2,n-1}}\ldots\overline{x_{2,0}}\underbrace{1\ldots1}_{n-1}) + (\underbrace{1\ldots1}_{2n-1}\overline{x_{3,pn}}) =$$

$$(\underbrace{1\ldots1}_{2n}) + (\underbrace{\overline{x_{2,n}}\,\overline{x_{2,n-1}}\ldots\overline{x_{2,0}}}_{n+1}\underbrace{1\ldots1}_{n-1}\overline{x_{3,pn}}) = (\underbrace{\overline{x_{2,n}}\,\overline{x_{2,n-1}}\ldots\overline{x_{2,0}}}_{n+1}\underbrace{1\ldots1}_{n-1}\overline{x_{3,pn}}) \quad (31)$$

By substituting above result in equation (17), the value of X is calculated as follow in which Yx_3 is calculated by concatenation of Y and x_3, so we don't need any additional hardware.

$$X = Y\,?\;\; 2^{pn+1} - 1) + x_3 \quad (32)$$

$$X = 2^{pn+1}Y - Y + x_3 = Yx_3 + \overline{Y} + 1 \quad (33)$$

Example:

For moduli set {2^n-1,2^n+1,2^{pn+1}-1}for p=2 and n=3 we have the moduli set {7,9,127}. We calculate X by residue representation (4,4,3) as follow:

x_1=4=$(100)_2$

x_2=4=$(0100)_2$

x_3=3=$(0000011)_2$

With respect to equations 23,24,29,31,33, we have:

v1=(010010)2=18

v2=(000010)2=2

v'=(101111)2=47

v4,2=(111100)2=60

Y=|18+2+47+60|63=|127|63=1

X=128×1-1+3=130

If we consider above results, we can see that the residue representation X=130, with respect to moduli set {7,9,127} is (4,4,3), which is truly calculated.

Hardware architecture of proposed reverse converter is shown in figure 1. For the hardware implementation we use modular adders and logic gates. In this structure the residue number (x_1,x_2,x_3) is changed to ((P/2)+4)-v_i vectors by operation preparation1(O.P.1), which is compose of (n+2)-bit not gates. We use a (2n)-bit CSA with EAC tree For calculating Y, in which first module adds the three v_1,v_2,v' vectors.

Since v_2 in equation (24) has n-1bits of "0", so n-1 F.As replace with n-1 H.As. v' in equation (31) contains n-2 bits of "1", so n-2 F.As replace n-2 XNOR/OR pairs. The other vectors also sequentially add with the result of previous module. The worst case is when we need (p/2)+1 addition modules and the best situation is when p/2 is a multiplication of three. In this case the delay will decrease. Here we have considered the worse scenario. The delay of each CSA is equal to the delay of one FA.

In the worst case, CSA tree needs (p/2)+1 CSA modules. Afterwards a (2n)-bit one's complement adder is to be added to the modulo 2^n-1, which is a CPA with EAC, therefore its delay is two times of the delay of a CPA that includes a 2n FA modules. In order to calculate the equation (33), we use 2n not gates and at the end, we use a ((2+p)n+1)-bit regular adder.

If we consider the delay of a CSA equal to a FA' delay and the delay of a n-bit CPA equal to an (2n)-bit F.A' delay, the final delay will be calculated as follow: delay=((6+p)n+(2+(p/2))) T_{FA}.

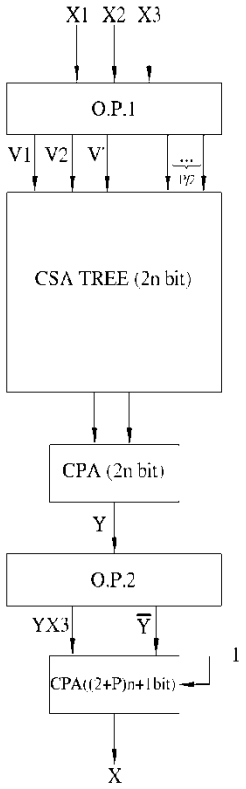

Figure 1. Reverse converter for general moduli set {2^n-1,2^n+1, 2^{pn+1}-1}

B. Reverse Converter Structure for Moduli Set {2^n-1,2^n+1, 2^{2n+1}-1}

General moduli set {2^n-1,2^n+1,2^{pn+1}-1}, for p=2 we have moduli set {2^n-1,2^n+1, 2^{2n+1}-1} which is considered as a special case.

$$X = Y?\ 2^{2n+1} - 1) + x_3 \tag{34}$$

$$v_1 = \left| 2^{n-1}(x_1 x_1) \right|_{2^{2n}-1} = (x_{1,0}x_{1,n-1}x_{1,n-2}...x_{1,0}x_{1,n-1}x_{1,n-2}...x_{1,1}) \tag{35}$$

$$v_2 = \left| 2^{2n-1}(0...0\ x_2) \right|_{2^{2n}-1} = (x_{2,0}\ 0...0\ x_{2,n}x_{2,n-1}...x_{2,1}) \tag{36}$$

$$v' = (\underbrace{\overline{x_{2,n}}\,\overline{x_{2,n-1}}...\overline{x_{2,0}}}_{n+1}\ \underbrace{1...1}_{n-1}\overline{x_{3,2n}}) \tag{37}$$

$$v_{4,2} = (\overline{x_{3,2n-1}}\,\overline{x_{3,2n-2}}...\overline{x_{3,0}}) \tag{38}$$

Finally X is calculated as follow :

$$X = 2^{2n+1}Y - Y + x_3 = Yx_3 + \overline{Y} + 1. \tag{39}$$

Example:

For moduli set {2^n-1,2^n+1,2^{2n+1}-1}for n=2 we have the moduli set {3,5,31}. We calculate X by residue representation (1,2,6) as follow:

$x_1=1=(01)_2$

$x_2=2=(010)_2$

$x_3=6=(00110)_2$

With respect to equations above, we have:

$v_1=(1010)_2=10$

$v_2=(0001)_2=1$

$v'=(1011)_2=11$

$v_{4,2}=(1001)_2=9$

$Y=|10+1+11+9|_{15}=|31|_{15}=1$

X=32×1-1+6=37

If we consider above results, we can see that the residue representation X=37, with respect to moduli set {3,5,31} is (1,2,6), which is truly calculated.

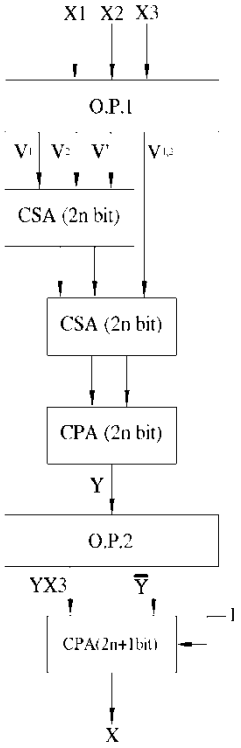

Figure 2. Reverse converter for moduli set {2^n-1,2^n+1, 2^{2n+1}-1}

The delay of this circuit is (8n+3) T_{FA}. Comparing this delay to the same form of moduli sets with the similar dynamic range, will show us that the delay is decreased. The details of the implementation are shown in figure 2.

TABLE.1 DELAY AND HARDWARE DETAILS OF REVERSE CONVERTER FOR DIFFERENT MODULI SETS.

Reverse Converter	Moduli Set	FA	NOT	XOR/ AND	XNOR /OR	Delay	DR
The Proposed Reverse Converter	{2^n-1,2^n+1,2^{2n+1}-1}	8n+4	5n+2	n-1	n-2	(8n+3) T_{FA} +2 T_{NOT}	4n+1
[16]	{$2^{n/2}$-1,$2^{n/2}$+1,2^n+1,2^{2n+1}-1}	8n+7	6n+4	2n-3	4n-3	(8n+4)T_{FA} +2 T_{NOT}	4n+1
[6]	{2^n-1,2^n-3,2^n+3,2^n+1}	26n+8	-	-	-	(7n+8)T_{FA} +2 ROM	4n
[11]	{2^n-1,2^n,2^{2n+1}-1}	9n+2	7n+2	5n+4	-	(7n+7)T_{FA}	4n+1
[8]	{2^n-1,2^n+1,2^{2n}+1}	8n+2	-	2n-2	-	(8n+2)T_{FA}	4n+1
[12]	{2^n-1,2^n,2^{2n-1}-1}	9n-2	7n-1	-	3n-3	(9n) T_{FA} + 3 T_{NOT}	4n-1

IV. PERFORMANCE EVALUATION

Dynamic range of proposed reverse converter for three general moduli set $\{2^n-1,2^n+1,2^{pn+1}-1\}$ is ((p+2)n+1)-bit and for the special case p=2 is (4n+1)-bit. The reverse converter of this moduli set is comparable to the reverse converters with similar dynamic range. The common and popular odd moduli sets with (4n+1)-bit dynamic range are $\{2^{n/2}-1,2^{n/2}+1,2^n+1,2^{2n+1}-1\}$ [16], $\{2^n-1,2^n-3,2^n+3,2^n+1\}$ [6]. In addition $\{2^n-1,2^n,2^{2n+1}-1\}$ [11], $\{2^n-1,2^n,2^{2n-1}-1\}$ [12] are two moduli sets with (4n+1)-bit dynamic range, which are compared to our reverse converter. Hardware implementation details and delay is estimated based on F.A.

Delay is listed in table 1. We can use the unit gate model [13], [15] to estimate the delay and the area. In this model each two input monotonic gate counts as one gate in the area and the delay. An XOR gate counts as two gates in the area and the delay and a F.A counts as seven gates in the area and four gates in the delay. Comparison between the delay and the area of reverse converters in this model are listed in table 2.

According to table 1 our reverse converter has better hardware cost and delay comparing to the reverse converters with moduli sets $\{2^{n/2}-1,2^{n/2}+1,2^n+1,2^{2n+1}-1\}$[16], $\{2^n-1,2^n,2^{2n-1}-1\}$[12]. Also it has better hardware cost in comparison with reverse converter with moduli set$\{2^n-1,2^n-3,2^n+3,2^n+1\}$ [6]. According to [1], using memory for large amount of n is not economically feasible for the delay and for the hardware. In addition two unusual moduli 2^n-3, 2^n+3, will cause decreasing of the performance in the arithmetic unit of RNS system. So from performance point of view our reverse converter is better than the reverse converter in [6] and It has less hardware costs comparing to the reverse converter with moduli set in $\{2^n-1,2^n,2^{2n+1}-1\}$ [11], although this module has less delay.

TABLE.2 AREA AND DELAY OF UNIT GATE OF REVERSE CONVERTER FOR DIFFERENT MODULI SETS.

Reverse Converter	Moduli Set	time-performance
The Proposed Reverse Converter	$\{2^n-1,2^n+1,2^{2n+1}-1\}$	$5+2\log_2^{(n+0.5)}$
[16]	$\{2^{n/2}-1,2^{n/2}+1,2^n+1,2^{2n+1}-1\}$	$5+2\log_2^{(n+0.5)}$
[6]	$\{2^n-1,2^n-3,2^n+3,2^n+1\}$	$7+2\log_2^{(n-1)}$
[11]	$\{2^n-1,2^n,2^{2n+1}-1\}$	$5+2\log_2^{(n+0.5)}$
[8]	$\{2^n-1,2^n+1,2^{2n}+1\}$	$8+2\log_2^{n}$
[12]	$\{2^n-1,2^n,2^{2n-1}-1\}$	$7+2\log_2^{(n-0.5)}$

There are some important parameters in designing an RNS like speed of internal RNS arithmetic processing. For estimating this parameter we use the method of [15] in which the time-performance is compared between moduli sets. The speed of arithmetic calculation for a moduli set is determined with slowest modulo which is the critical modulo. We use the unit gate delay of parallel prefix adder for critical modulo of moduli sets of table 1, and the results are shown in table 3. The moduli set $\{2^n-1,2^n+1,2^{2n}+1\}$[8], with (4n)-bit odd dynamic range, with respect to table 1 has better delay and hardware cost comparing to our reverse converter with 4n+1-bit dynamic range. however, because of two moduli in form of 2^n+1 we will have a decrease in performance of RNS arithmetic unit. So time-performance of our reverse converter is better than it. So we can conclude that the proposed reverse converter in this paper is better than the other reverse converter with similar dynamic range.

TABLE.3 COMPARISON BETWEEN TIME-PERFORMANCE OF DIFFERENT MODULI SETS.

Reverse Converter	Moduli Set	Area	Delay
The Proposed Reverse Converter	$\{2^n-1,2^n+1,2^{2n+1}-1\}$	67n+21	43n+9
[16]	$\{2^{n/2}-1,2^{n/2}+1,2^n+1,2^{2n+1}-1\}$	80n+35	56n+14
[11]	$\{2^n-1,2^n,2^{2n+1}-1\}$	85n+28	58n+22
[8]	$\{2^n-1,2^n+1,2^{2n}+1\}$	62n+8	38n+2
[12]	$\{2^n-1,2^n,2^{2n-1}-1\}$	79n+24	52n-18

V. CONCLUSION

In this paper a general three moduli set for even p and it's reverse converter is proposed. This moduli set with ((2+p)n+1)-bit variant dynamic range can have different dynamic range according to different applications. The odd moduli set leads to efficient implementation of internal circuits for fundamental problems in RNS arithmetic and in overall RNS system. The reverse converter in this paper has a better performance in hardware cost and delay comparing to the other reverse converters with similar dynamic range.

REFERENCES

[1] A. Omondi, B. Premkumar. Residue Number Systems: Theory and Implementation. London: Imperial College Press, 2007.

[2] B. Parhami. Computer Arithmetic: Algorithms and Hardware Designs. New York: Oxford University Press, 2000.

[3] G. C. Cardarilli, A. Nannarelli, M. Re. Residue Number System for Low-Power Dsp Applications. In Proc. 41nd Asilomar Conf. Signals Syst. Comput, 2007, 1412–1416.

[4] R. Conway, J. Nelson. Improved RNS Fir Filter Architectures. IEEE Trans. Circuits Syst. Exp. Briefs, 2004, 51(1):26–28.

[5] W. Wang, M. N. S. Swamy, M. O. Ahmad. RNS Application for Digital Image Processing. In Proc. 4th IEEE Int. Workshop System-On-Chip for Real Time Appl.. 2004:77–80.

[6] M. H. Sheu, S. H. Lin, C. Chen, S. W. Yang. An Efficient VLSI Design For A Residue To Binary Converter For General Balance Moduli {2^n-1,2^n-3,2^n+3,2^n+1}. IEEE Transactions Circuits System II: Express Briefs, 2004, 51(2):152-155.

[7] W. Zhang , P. Siy. An Efficient Design Of Residue to Binary Converter for Four Moduli Set (2^n–1,2^n+1, 2^{2n}–2, 2^{2n+1}–3) Based on New Crt II. Elsevier Journal of Information Sciences, 2008, 178(1):264-279.

[8] W. Wang, M.N.S Swamy, M.O. Ahmad, Y.Wang. A Study of the Residue to Binary Converter for the Three-Moduli Sets. IEEE Transactions on Circuits and Systems I, 2003, 50(2): 235-243.

[9] M.A. Shang, H.U. Jianhao. An Efficient RNS Parity Checker for Moduli Set {2^n-1, 2^n+1, 2^{2n}+1} and Its Applications. Sci China Ser F-Inf Sci, 2008, 51(10):1563-1571.

[10] Y.Wang, X. Song, M. Aboulhamid, H. Shen. Adder Based Residue to Binary Numbers Converters for {2^n-1, 2^n, 2^n+1}. IEEE Trans. Signal Process, 2002, 50(7):1772–1779.

[11] K. A. Gbolagade, R. Chaves, L. Sousa, S.D. Cotofana. An Improved RNS Reverse Converter for the {2^{2n+1}−1,2^n,2^n−1} Moduli Set. IEEE International Symposium on Circuits and Systems, 2010:2103-2106.

[12] A. S. Molahosseini, S. Sezavar, K. Navi. A New Design of Reverse Converter for a Three-Moduli Set. International Symposium on Intelligent Signal Processing and Communication Systems, 2009: 57-60.

[13] L. Kalampoukas, D. Nikolos, C. Efstathiou. High-Speed Parallel-Perfix Modulo 2^n-1 Adders. IEEE Transactions on Computers, 2000: 673- 679.

[14] R. Zimmermann. Efficient VlSI Implementation of Modulo (2^n+1) Addition and Multiplication. Proceedings of the 14th Symposium on Computer Arithmetic, IEEE, 1999:158-167.

[15] K. Yuan-Ching, L. Su-Hon, S. Ming-Hwa, W. Jia-You. Efficient VlSI Design of a Reverse RNS Converter for New Flexible 4-Moduli Set (2^{p+K},2^p+1,2^p−1,2^{2p}+1). IEEE International Symposium on Circuits and Systems, Iscas 2009:437 – 440.

[16] A. S. Molahosseini, K. Navi, F. Teymouri. A New Four-Modulus RNS to Binary Converter. IEEE International Symposium on Circuits and Systems, 2010: 4161-4164.

[17] A. S. Molahosseini, K. Navi, O. Hashemipur, A. Jalali. An Efficient Architecture for Designing Reverse Converters Based on A General Three-Moduli Set. Elsevier Journal of Systems Architecture, 2008, 54(10):929-934.

[18] P.V. Ananda Mohan. New Reverse Converters for The Moduli Set {2^n-3,2^n-1,2^n+1,2^n+3}. Elsevier Jurnal of Aeu - International Journal of Electronics and Communications, 2008, 62(9):643-658.

[19] M. Hosseinzadeh, A. Sabbagh, K. Navi. A Fully Parallel Reverse Converter. International Journal of Electrical, Computer and Systems Engineering, 2007, 1(3):183-187.

[20] K. Navi, A. S. Molahosseini, M. Esmaeildoust. How to Teach Residue Number System to Computer Scientists and Engineers. IEEE Transactions on Education 1,2010.

[21] B. Cao, T. Srikanthan, C.H. Chang. Efficient Reverse Converters for the 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}. IEE Proc Comput Digit Tech, 2005, 152:687–96.

[22] A. Dhurkadas. A High Speed Realisation of a Residue to Binary Number System Converter. IEEE Transactions on Cas, Part II, 1998, 45: 446-447.

[23] P.V. Ananda Mohan, A.B. Premkumar. 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}. IEEE Transactions on Cas, 2007, Part I, 54: 1245-1254.

Mehdi Hosseinzadeh Received B.Sc. in Computer Hardware Engineering from Islamic Azad University, Dezful branch, Iran in 2003. He also received the M.Sc. and Ph.D. degrees in Computer System Architecture from the Science and Research Branch, Islamic Azad University, Tehran, Iran in 2005 and 2008, respectively. He is currently Assistant Professor in Department of Computer Engineering of Science and Research Branch of Islamic Azad University, Tehran, Iran. His research interests are Computer Arithmetic with emphasis on Residue Number System, Cryptography, Network Security and E-Commerce.

Keihaneh kia received the B.Sc. degree (with highest honors) from Shahid Bahonar University of Kerman, Kerman, Iran, in 2007, and the M.Sc. degree from Islamic Azad University (IAU), Science and Research Branch, Tehran, Iran, in 2010, both in computer engineering. Her research interests include residue number System and carbon nanotubes.

