**This document is downloaded from DR-NTU (https://dr.ntu.edu.sg)
Nanyang Technological University, Singapore.**

An efficient reverse converter for the 4-moduli set {2^n -1, 2^n, 2^n + 1, 2^2n + 1} based on the new Chinese remainder theorem

Cao, Bin; Chang, Chip Hong; Srikanthan, Thambipillai

2003

Cao, B., Chang, C. H., & Srikanthan, T. (2003). An efficient reverse converter for the 4-moduli set {2^n -1, 2^n, 2^n + 1, 2^2n + 1} based on the new Chinese remainder theorem. IEEE Transactions on Circuits And Systems-I: Fundamental Theory and Applications, 50(10), 1296-1303.

https://hdl.handle.net/10356/91437

https://doi.org/10.1109/TCSI.2003.817789

IEEE Transactions on Circuits And Systems-I: Fundamental Theory and Applications © 2003 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE. This material is presented to ensure timely dissemination of scholarly and technical work. Copyright and all rights therein are retained by authors or by other copyright holders. All persons copying this information are expected to adhere to the terms and constraints invoked by each author's copyright. In most cases, these works may not be reposted without the explicit permission of the copyright holder. http://www.ieee.org/portal/site.

Downloaded on 26 Aug 2022 00:31:23 SGT

An Efficient Reverse Converter for the 4-Moduli Set $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ Based on the New Chinese Remainder Theorem

Bin Cao, Chip-Hong Chang, *Senior Member, IEEE*, and Thambipillai Srikanthan, *Senior Member, IEEE*

***Abstract*—The inherent properties of carry-free operations, parallelism and fault-tolerance have made the residue number system a promising candidate for high-speed arithmetic and specialized high-precision digital signal-processing applications. However, the reverse conversion from the residues to the weighted binary number has long been the performance bottleneck, particularly when the number of moduli set increases beyond 3. In this paper, we present an elegant residue-to-binary conversion algorithm for a new 4-moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$. The new Chinese remainder theorem introduced recently has been employed to exploit the special properties of the proposed moduli set where modulo corrections are done without resorting to the costly and time consuming modulo operations. The resulting architecture is notably simple and can be realized in hardware with only bit reorientation and one multioperand modular adder. The new reverse converter has superior area-time complexity in comparison with the reverse converters for several other 4-moduli sets.**

***Index Terms*—New Chinese remainder theorem (CRT), residue arithmetic, residue number system (RNS), residue-to-binary converter.**

I. Introduction

THE RESIDUE number system (RNS) is a means of representing number in a nonweighted form in order to heighten the parallelism and modularity that can be exploited by the very large-scale integration (VLSI) technology [1], [2]. By decomposing large binary numbers into smaller residues, addition and subtraction in RNS arithmetic have no inter-digit carries or borrows, and multiplication can be performed without the need to generate partial products. With the ever increasing bus width for high-precision computation, and the demand for digital signal processors to deliver reliable and giga operations per second (GOPS)-like performance with microwatts-like power budget, the elimination of carry chain coupled with the advantages offered by VLSI fabrication technology makes RNS-based arithmetic attractive for many important applications, especially those that have to deal with a lot of high-speed vector processing [1]–[5]. As the peripheral interfaces of most digital systems are still based on the weighted number system, the overhead incurred in the conversions into and out of the RNS has been the major critique limiting the diffusion of RNS-based processors. It is well acknowledged that the forward conversion from the binary number to residues is conceivably simple and efficient architectures have been devised for residue generator based on special moduli of the 2^n-*type* (i.e., expressible in the form 2^n or $2^n \pm 1$) [6]. However, the reverse conversion from residue-to-binary number presents a significantly higher hurdle that offsets the performance gained in the RNS. During the past several decades, the reverse conversion algorithms are based primarily on the Chinese remainder theorem (CRT) [7]–[10] or mixed-radix conversion (MRC) [11], [12]. The use of CRT often involves a large modulo M adder where M is the product of all moduli, whereas MRC is a sequential process that often requires a number of lookup tables. Both of them are not efficient for practical systems with large dynamic range.

Special moduli sets have been used extensively to reduce the hardware complexity in the implementation of reverse converters. Among which the triple moduli set, $\{2^n - 1, 2^n, 2^n + 1\}$ has gained unprecedented popularity by feat of its inherent number theoretic properties in the CRT algorithm. Several researchers have proposed different kinds of residue-to-binary converters using CRT. Andraos and Ahmad [13] derived the closed-form expressions of the moduli inverses, and used them to reduce the conversion complexity. Carry save adders (CSAs) with end-around carry (EAC) were introduced by Piestrak [14] to allow a very efficient implementation of the residue-to-binary converters based on CRT. Recently, Wang [15], [16] proposed the revolutionary New CRT theorems, which have inherited the merits of the classical CRT and MRC algorithms. With general moduli sets, the reverse converters based on the New CRT are still relatively complex. On the other hand, if the New CRT I and II are applied to the special triple moduli set, the algorithm of the reverse conversion can be simplified, and reduced hardware complexity and improved speed are achieved compared with the converters designed with the celebrated traditional CRT [17].

Inspired by the performance gain of the reverse converter for triple moduli set and the potential simplification made possible by the number theoretic interaction of the New CRT, we explore new moduli set which would take advantage of the New CRT to increase the parallelism and extend the dynamic range of the RNS. We have derived a new 4-moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ that retains the efficient forward conversion and

Manuscript received August 23, 2002; revised March 28, 2003. This paper was recommended by Associate Editor C.-W. Wu.

B. Cao and T. Srikanthan are with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: caobin@pmail.ntu.edu.sg; astsrikan@ntu.edu.sg).

C.-H. Chang is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: echchang@ntu.edu.sg).

Digital Object Identifier 10.1109/TCSI.2003.817789

1057-7122/03$17.00 © 2003 IEEE

residue arithmetic. Its ensuing residue-to-binary converter designed using the New CRT is as efficient as that of the triple moduli set, while the dynamic range has raised from $3n-1$ bits to $5n-1$ bits. Techniques exploiting the special properties of power of two modulo (2^p-1) are employed to simplify the final modulo correction and allow for an efficient circuit realization of a memoryless reverse converter. One of the newest reverse converters based on the 4-moduli set was proposed by Bhardwaj *et al.* [18] which used the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$ for odd number n. The dynamic range is $4n+1$ bits and the converter is based on linear ROM and modular adders. The total delay is approximately 10 $n+3$ full adder (FA) delays, and the area cost is quadratic in n. An improved memoryless reverse converter for 4-moduli set of $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ with an overall delay of $14n+8$ FA delays was recently proposed by Vinod and Premkumar [19]. This 4-moduli set is valid only for even values of n, which restricts its dynamic range further.

The remaining sections of this paper are organized as follows. In Section II, an overview of RNSs and the new CRT theorems is provided. In Section III, the new 4-moduli set is proposed. The proof of its relatively prime postulation is established followed by the derivation of a novel algorithm for the reverse conversion based on the new CRT. Its hardware implementation is presented in Section IV. The architectural cost and performance of the proposed reverse converter are evaluated and compared with other converters in Section V, before the conclusion.

II. Background

In an RNS, an integer X can be represented by an n-tuple of residues, $(x_1, x_2, \ldots, x_n)$ defined over a set of relatively prime moduli $\{P_1, P_2, \ldots, P_n\}$, where $gcd(P_i, P_j) = 1$ for $1 \leq i, j \leq n$, and $i \neq j$. The residue x_i is a smaller weighted binary number expressible as

$$x_i = |X|_{P_i} = \sum_{j=0}^{\lceil \log_2 P_i \rceil - 1} b_j 2^j \qquad (1)$$

where $|X|_{P_i}$ is a simplified notation for X mod P_i. The term forward and reverse conversions are commonly used to describe the conversions of the weighted binary representation X to and from its residue set $(x_1, x_2, \ldots, x_n)$, respectively. The bottleneck operation in a RNS is the reverse conversion, which can be calculated using the CRT [1] as follows:

$$X = \left| \sum_{i=1}^{n} \left| \frac{x_i}{\hat{M}_i} \right|_{P_i} \cdot \hat{M}_i \right|_M \qquad (2)$$

where $M = \prod_{i=1}^{n} P_i$, and $\hat{M}_i = M/P_i$.

The weighted binary number X can be also calculated by the new CRT-I [15]

$$\begin{aligned} X =& x_1 \\ &+ P_1 \left| \begin{matrix} k_1(x_2-x_1) + k_2P_2(x_3-x_2) + \cdots \\ +k_{n-1}P_2P_3 \cdots P_{n-1}(x_n-x_{n-1}) \end{matrix} \right|_{P_2P_3 \cdots P_{n-1}P_n} \end{aligned} \qquad (3)$$

where

$$\begin{aligned} |k_1P_1|_{P_2P_3 \cdots P_n} &= 1 \\ |k_2P_1P_2|_{P_3P_4 \cdots P_n} &= 1 \\ &\cdots \\ |k_{n-1}P_1P_2 \cdots P_{n-1}|_{P_n} &= 1. \end{aligned} \qquad (4)$$

The product of all moduli, M, is called the dynamic range. For any integer X within the dynamic range, i.e., $0 \leq X < M$, its residue representation is canonical. The dynamic range of an RNS is also expressible in d bits, where $d = \lfloor \log_2 M \rfloor$. Therefore, the RNS is said to have a dynamic range of $3n-1$ bits with the moduli set $\{2^n-1, 2^n, 2^n+1\}$, $4n+1$ (for $n = 2k+1$) bits with the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n+1}+1\}$, and $5n-1$, with the moduli set $\{2^n-1, 2^n, 2^n+1\ 2^{2n}+1\}$.

In what follows, we will establish the number theoretic properties of the proposed 4-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ and derive the closed-form expressions for the inverses under the New CRT-I that forms the basis of our algorithm for the reverse converter.

III. Reverse Converter for 4-Moduli Set $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$

Before we can apply the New CRT-I to derive the algorithm for the reverse converter of the proposed 4-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$, we need to prove that these moduli are in fact pairwise relatively prime for the validity of the RNS. The following theorem is useful for this purpose.

Theorem 1: For natural number a, b, and c, if $a|b$ and $a|(b+c)$, then $a|c$.

Proof: If $a|b$, and $a|(b+c)$, then $b = ka$, and $b+c = ma$ for some integers k and m where $k \geq 1$ and $m > 1$. Since $b+c > b$, $m > k$. Let the nature number $n = m-k$, then $c = (b+c)-b = ma-ka = (m-k)a = na$. Therefore, $a|c$.

Theorem 2: The moduli from the 4-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ are pairwise relatively prime for any natural number n.

Proof: It has been established that the moduli 2^n-1, 2^n and 2^n+1 are pairwise relatively prime [1]. Hence, we only need to prove that $2^{2n}+1$ is relatively prime to the other moduli.

Assume $2^{2n}+1$ is divisible by 2^n, then $2^{2n}+1 = k \cdot 2^n$, for some natural number $k > 2^n$. Rearranging the above equation, we have $k \cdot 2^n - 2^{2n} = 1 \Rightarrow 2^n \cdot (k-2^n) = 1$. Since $k > 2^n \Rightarrow k - 2^n > 0$ and 2^n is an even natural number, the identity $2^n \cdot (k-1^n) = 1$ is invalid. By contradiction, $2^{2n}+1$ is not divisible by 2. Hence, $2^{2n}+1$ is prime to 2^n.

Assume that $2^{2n}+1$ is not prime to 2^n+1, there exists a natural number $a \geq 2$ such that $a = gcd(2^n+1, 2^{2n}+1)$. Thus, we have $a|2^{2n}+1$ and $a|2^n+1$.

Since $2^{2n}+1 = 2^n(2^n-1)+2^n+1$, $a|2^{2n}+1$ implies that $a|(2^n(2^n-1)+2^n+1)$. According to *Theorem 1* and the assumption, $a|2^n(2^n-1)$. The divisibility property [20] suggests that either i) $a|2^n$ or ii) $a|2^n-1$. i) If $a|2^n$, based on the above assumption that $a|2^n+1$, a is the common divisor of 2^n+1 and 2^n. This contradicts the well-established fact that 2^n+1 is relatively prime to 2^n. ii) if $a|2^n-1$, based on the assumption, $a|2^n+1$ and a is the common divisor of 2^n+1 and 2^n-1. This is, again, a contradiction to the proven fact that

2^n+1 is relatively prime to 2^n-1. Since both cases cannot be established, the assumption that 2^n+1 is not prime to $2^{2n}+1$ is invalid. Therefore, 2^n+1 is relatively prime to $2^{2n}+1$.

Assume that $2^{2n}+1$ is not prime to 2^n-1, there exists a natural number $a \geq 2$ such that $a = gcd(2^n-1, 2^{2n}+1)$. So, we have $a|2^{2n}+1\ |2^n-1$.

Since $2^{2n}+1 = 2^n\ (2^n-1)+2^n+1$, $a|2^{2n}+1$ implies that $a|(2^n(2^n-1)+2^n+1)$. According to *Theorem 1* and the assumption, we have $a|2^n+1$, so a is the common divisor of 2^n+1 and 2^n-1. This is in contradiction with the fact that 2^n-1 and 2^n+1 are relatively prime. Hence, we rule out the assumption and conclude that $2^{2n}+1$ is prime to 2^n-1.

Since $2^{2n}+1$ is pairwise relatively prime to all the other moduli in the set $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$, the relative primality of the proposed 4-moduli set is established.

With the 4-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$, the reverse conversion algorithm from the residues (x_1, x_2, x_3, x_4) to the weighted binary number X is derived in the sequel based on the New CRT-I.

Let $P_1 = 2^n$, $P_2 = 2^n+1$, $P_3 = 2^{2n}+1$ and $P_4 = 2^n-1$. By (4), we have

$$|k_1 \cdot 2^n|_{(2^n+1)(2^{2n}+1)(2^n-1)} = 1 \tag{5}$$

$$|k_2 \cdot 2^n\,(2^n+1)|_{(2^n-1)(2^{2n}+1)} = 1 \tag{6}$$

$$\left|k_3 \cdot 2^n\left(2^{2n}+1\right)(2^n+1)\right|_{2^n-1} = 1. \tag{7}$$

The three multiplicative inverses k_1, k_2, and k_3 are given as follows:

$$k_1 = 2^{3n} \tag{8a}$$

$$k_2 = 2^{3n-2}+2^{2n-1}-2^{n-2} \tag{8b}$$

$$k_3 = 2^{n-2}. \tag{8c}$$

Proof of (8a):

$$\begin{aligned}|k_1 \cdot 2^n|_{(2^n+1)(2^{2n}+1)(2^n-1)} &= \left|2^{3n} \cdot 2^n\right|_{2^{4n}-1}\\ &= \left|2^{4n}\right|_{2^{4n}-1}\\ &= 1.\end{aligned}$$

Proof of (8b):

$$\begin{aligned}&|k_2 \cdot 2^n\,(2^n+1)|_{(2^n-1)(2^{2n}+1)}\\ &\quad= \left|\left(2^{3n-2}+2^{2n-1}-2^{n-2}\right) \cdot 2^n\,(2^n+1)\right|_{2^{3n}-2^{2n}+2^n-1}\\ &\quad= \left|2^{5n-2}+2^{4n-1}-2^{3n-2}+2^{4n-2}\right.\\ &\qquad\quad \left.+\ 2^{3n-1}-2^{2n-2}\right|_{2^{3n}-2^{2n}+2^n-1}\\ &\quad= \left|\begin{array}{l}-2^{2n-2}\left(2^{3n}-2^{2n}+2^n-1\right)\\ +2^{5n-2}+2^{4n-1}-2^{3n-2}+2^{4n-2}\\ +2^{3n-1}-2^{2n-2}\end{array}\right|_{2^{3n}-2^{2n}+2^n-1}\\ &\quad= \left|2^{4n}\right|_{2^{3n}-2^{2n}+2^n-1}\\ &\quad= \left|2^n\left(2^{3n}-2^{2n}+2^n-1\right)\right.\\ &\qquad\quad \left.+\ 2^{3n}-2^{2n}+2^n\right|_{2^{3n}-2^{2n}+2^n-1}\\ &\quad= \left|2^{3n}-2^{2n}+2^n\right|_{2^{3n}-2^{2n}+2^n-1}\\ &\quad= 1.\end{aligned}$$

Proof of (8c): It is trivial to show that $|2^n|_{2^n-1} = 1$. Using the identities $|a+b|_c = ||a|_c+|b|_c|_c$ and $|ab|_c = ||a|_c|b|_c|_c$, we have $|2^n+1|_{2^n-1} = 2$, $|2^{2n}+1|_{2^n-1} = |2^n \cdot 2^n+1|_{2^n-1} = 2$. Thus

$$\begin{aligned}&\left|k_3 \cdot 2^n\left(2^{2n}+1\right)(2^n+1)\right|_{2^n-1}\\ &\quad= \left|2^{n-2} \cdot 2^n\left(2^{2n}+1\right)(2^n+1)\right|_{2^n-1}\\ &\quad= \left|2^{n-2} \cdot 1 \cdot 2 \cdot 2\right|_{2^n-1}\\ &\quad= |2^n|_{2^n-1}\\ &\quad= 1.\end{aligned}$$

Theorem 3: In an RNS defined by the 4-moduli set $\{2^n-1, 2^n, 2^n+1$, and $2^{2n}+1\}$, the weighted binary number X can be calculated from the residues (x_1, x_2, x_3, and x_4) by

$$X = x_2 + 2^n \cdot |A \cdot x_1 + B \cdot x_2 + C \cdot x_3 + D \cdot x_4|_{2^{4n}-1} \tag{9}$$

where

$$A = 2^{n-2}\left(2^{3n}+2^{2n}+2^n+1\right) \tag{10a}$$

$$B = -2^{3n} \tag{10b}$$

$$C = 2^{3n}-(2^n+1)\left(2^{3n-2}+2^{2n-1}-2^{n-2}\right) \tag{10c}$$

$$\begin{aligned}D = {}&(2^n+1)\left(2^{3n-2}+2^{2n-1}-2^{n-2}\right)\\ &-2^{n-2}\,(2^n+1)\left(2^{2n}+1\right).\end{aligned} \tag{10d}$$

Proof: By substituting $P_1 = 2^n$, $P_2 = 2^n+1$, $P_3 = 2^{2n}+1$, $P_4 = 2^n-1$, and the values of k_1 to k_3 from (8a)–(8c) into (3), we have

$$X = x_1 + 2^n \left|\begin{array}{l}2^{3n}\,(x_3-x_2)\\ +\left(2^{3n-2}+2^{2n-1}-2^{n-2}\right)\\ \cdot\,(2^n+1)\,(x_4-x_3)\\ +2^{n-2}\,(2^n+1)\\ \cdot\left(2^{2n}+1\right)(x_1-x_4)\end{array}\right|_{2^{4n}-1}.$$

It should be noted that the residues corresponding to P_1, P$_2$, P_3 and P_4 are, respectively, x_2, x_3, x_4, and x_1 of (3). The result follows directly by expanding the terms and grouping the coefficients of x_1, x_2, x_3 and x_4.

IV. Hardware Realization of Reverse Converter

We use *Theorem 3* as the basis to devise a VLSI efficient reverse converter. Before developing it into a hardware architecture, the closed-form expressions of (9) and (10) can be further simplified to reduce the hardware complexity by exploiting the special properties of the modular M operation.

Let the residues x_1, x_2, x_3, and x_4 be represented by binary strings of different length

$$\begin{aligned}x_1 &= (X_{1,\,n-1}X_{1,\,n-2}\cdots X_{1,\,1}X_{1,\,0})_2\\ x_2 &= (X_{2,\,n-1}X_{2,\,n-2}\cdots X_{2,\,1}X_{2,\,0})_2\\ x_3 &= (X_{3,\,n}X_{3,\,n-1}\cdots X_{3,\,1}X_{3,\,0})_2\\ x_4 &= (X_{4,\,2n}X_{4,\,2n-1}\cdots X_{4,\,1}X_{4,\,0})_2\,.\end{aligned}$$

Furthermore, let

$$Y = \left|\sum_{i=1}^{4}\beta_i\right|_{2^{4n}-1}$$

where

$$\begin{aligned}\beta_1 &= |A \cdot x_1|_{2^{4n}-1}\\ \beta_2 &= |B \cdot x_2|_{2^{4n}-1}\\ \beta_3 &= |C \cdot x_3|_{2^{4n}-1}\\ \beta_4 &= |D \cdot x_4|_{2^{4n}-1}.\end{aligned}$$

According to *Theorem 3*, X can be calculated by $X = x_2 + 2^n \cdot Y$.

The following property can be used to evaluate Y.

Property 1: Multiplying an integer, x by 2^r modulo $(2^p - 1)$ can be accomplished by expressing x in a p bits binary representation and then shifting it circularly by r bits to the left.

Property 1 is proposed in [1]. As an example, let $x = 6, p = 5$ and $r = 3$, then, we have

$$|x \cdot 2^r|_{2^p-1} = \left|6 \cdot 2^3\right|_{2^5-1} = CLS(00110, 3) = (10001)_2 = 17$$

where the function $CLS(x, r)$ is used to denote a circular shift of the binary number x by r bits to the left. In the sequel, we apply *Property 1* recursively to replace β_i by the arrangement of bits from the residues interleaving with strings of binary constants "0" and "1."

Evaluating β_1

$$\begin{aligned}\beta_1 &= \left|2^{n-2}\left(2^{3n} + 2^{2n} + 2^n + 1\right) \cdot x_1\right|_{2^{4n}-1}\\ &= |CLS\left\{\left(CLS\left(x_1, 3n\right) + CLS\left(x_1, 2n\right)\right.\right.\\ &\qquad \left.\left. + CLS\left(x_1, n\right) + x_1\right), n-2\right\}|_{2^{4n}-1}\\ &= CLS\left(\underbrace{X_{1, n-1} \cdots X_{1, 0}}_{n}\underbrace{X_{1, n-1} \cdots X_{1, 0}}_{n}\right.\\ &\qquad \left.\underbrace{X_{1, n-1} \cdots X_{1, 0}}_{n}\underbrace{X_{1, n-1} \cdots X_{1, 0}}_{n}, n-2\right)\\ &= \underbrace{X_{1, 1}X_{1, 0}}_{2}\underbrace{X_{1, n-1} \cdots X_{1, 0}}_{n}\underbrace{X_{1, n-1} \cdots X_{1, 0}}_{n}\\ &\qquad \underbrace{X_{1, n-1} \cdots X_{1, 0}}_{n}\underbrace{X_{1, n-1} \cdots X_{1, 2}}_{n-2}.\end{aligned} \tag{11}$$

Evaluating β_2

$$\begin{aligned}\beta_2 &= \left|-2^{3n} \cdot x_2\right|_{2^{4n}-1}\\ &= \left|2^{4n} - 1 - CLS\left(x_2, 3n\right)\right|_{2^{4n}-1}\\ &= \underbrace{\overline{X}_{2, n-1} \cdots \overline{X}_{2, 0}}_{n}\underbrace{11 \cdots 11}_{3n}.\end{aligned} \tag{12}$$

Evaluating β_3

$$\begin{aligned}\beta_3 &= |C \cdot x_3|_{2^{4n}-1}\\ &= \left|\left\{2^{3n} - \left(2^{4n-2} + 2^{3n-1} - 2^{2n-2} + 2^{3n-2}\right.\right.\right.\\ &\qquad \left.\left.\left. + 2^{2n-1} - 2^{n-2}\right)\right\} \cdot x_3\right|_{2^{4n}-1}\\ &= \left|\left\{-\left(2^{4n-2} + 2^{2n-2}\right) + \left(2^{3n-2} + 2^{n-2}\right)\right\} \cdot x_3\right|_{2^{4n}-1}.\end{aligned}$$

Define

$$\begin{aligned}\beta_{3, 1} &= \left|-\left(2^{4n-2} + 2^{2n-2}\right) \cdot x_3\right|_{2^{4n}-1}\\ \beta_{3, 2} &= \left|\left(2^{3n-2} + 2^{n-2}\right) \cdot x_3\right|_{2^{4n}-1}\end{aligned}$$

so that $\beta_3 = |\beta_{3, 1} + \beta_{3, 2}|_{2^{4n}-1}$.

Now, $\beta_{3, 1}$ and $\beta_{3, 2}$ can be evaluated by recursive applications of *Property 1*

$$\begin{aligned}\beta_{3, 1} &= \left|-\left(2^{4n-2} + 2^{2n-2}\right) \cdot x_3\right|_{2^{4n}-1}\\ &= \left|-2^{2n-2}\left(2^{2n} + 1\right) \cdot x_3\right|_{2^{4n}-1}\\ &= |-CLS\left(x_3 + CLS(x_3, 2n), 2n-2\right)|_{2^{4n}-1}\end{aligned}$$

where

$$\begin{aligned}&x_3 + CLS\left(x_3, 2n\right)\\ &\quad = \underbrace{0 \cdots 0}_{n-1}\underbrace{X_{3, n} \cdots X_{3, 0}}_{n+1}\underbrace{0 \cdots 0}_{n-1}\underbrace{X_{3, n} \cdots X_{3, 0}}_{n+1}\\ &CLS\left(x_3 + CLS(x_3, 2n), 2n-2\right)\\ &\quad = X_{3, 1}X_{3, 0}\underbrace{0..0}_{n-1}\underbrace{X_{3, n} \cdots X_{3, 0}}_{n+1}\underbrace{0..0}_{n-1}\underbrace{X_{3, n} \cdots X_{3, 2}}_{n-1}.\end{aligned}$$

Eventually, we have

$$\begin{aligned}\beta_{3, 1} &= |-CLS\left(x_3 + CLS(x_3, 2n), 2n-2\right)|_{2^{4n}-1}\\ &= \left|2^{4n} - 1 - CLS\left(x_3 + CLS(x_3, 2n), 2n-2\right)\right|_{2^{4n}-1}\\ &= \underbrace{\overline{X}_{3, 1}\overline{X}_{3, 0}}_{2}\underbrace{1 \cdots 1}_{n-1}\underbrace{\overline{X}_{3, n} \cdots \overline{X}_{3, 0}}_{n+1}\underbrace{1 \cdots 1}_{n-1}\underbrace{\overline{X}_{3, n} \cdots \overline{X}_{3, 2}}_{n-1}\end{aligned} \tag{13}$$

$$\begin{aligned}\beta_{3, 2} &= \left|\left(2^{3n-2} + 2^{n-2}\right) \cdot x_3\right|_{2^{4n}-1}\\ &= \left|2^{n-2}\left(2^{2n} + 1\right) \cdot x_3\right|_{2^{4n}-1}\\ &= CLS\left(x_3 + CLS\left(x_3, 2n\right), n-2\right)\\ &= 0\underbrace{X_{3, n} \cdots X_{3, 0}}_{n+1}\underbrace{0 \cdots 0}_{n-1}\underbrace{X_{3, n} \cdots X_{3, 0}}_{n+1}\underbrace{0 \cdots 0}_{n-2}.\end{aligned} \tag{14}$$

Evaluating β_4

$$\begin{aligned}\beta_4 &= \left|\left\{\begin{array}{l}\left(2^n + 1\right)\left(2^{3n-2} + 2^{2n-1} - 2^{n-2}\right)\\ -2^{n-2}\left(2^n + 1\right)\left(2^{2n} + 1\right)\end{array}\right\} \cdot x_4\right|_{2^{4n}-1}\\ &= \left|\left(2^{3n-1} - 2^{n-1}\right) \cdot x_4\right|_{2^{4n}-1} = |\beta_{4, 1} + \beta_{4, 2}|_{2^{4n}-1}\end{aligned}$$

where

$$\begin{aligned}\beta_{4, 1} &= \left|2^{3n-1} \cdot x_4\right|_{2^{4n}-1}\\ &= CLS\left(x_4, 3n-1\right)\\ &= \underbrace{X_{4,n}X_{4,n-1} \cdots X_{4,0}}_{n+1}\underbrace{0 \cdots 0}_{2n-1}\underbrace{X_{4,2n}X_{4,2n-1} \cdots X_{4,n+1}}_{n}\end{aligned} \tag{15}$$

$$\begin{aligned}\beta_{4,2} &= \left|-2^{n-1} \cdot x_4\right|_{2^{4n}-1}\\ &= \left|2^{4n} - 1 - CLS\left(x_4, n-1\right)\right|_{2^{4n}-1}\\ &= \underbrace{1 \cdots 1}_{n}\underbrace{\overline{X}_{4,2n} \cdots \overline{X}_{4,0}}_{2n+1}\underbrace{1 \cdots 1}_{n-1}.\end{aligned} \tag{16}$$

Now Y can be expressed as the sum of the binary strings given by (11)–(16)

$$\begin{aligned}Y &= \left|\sum_{i=1}^{4}\beta_i\right|_{2^{4n}-1}\\ &= |\beta_1 + \beta_2 + \beta_{3, 1} + \beta_{3, 2} + \beta_{4, 1} + \beta_{4, 2}|_{2^{4n}-1}.\end{aligned} \tag{17}$$

From (9) and (17), it can be seen that the calculation of X is magically simple and elegant. It involves only multioperand

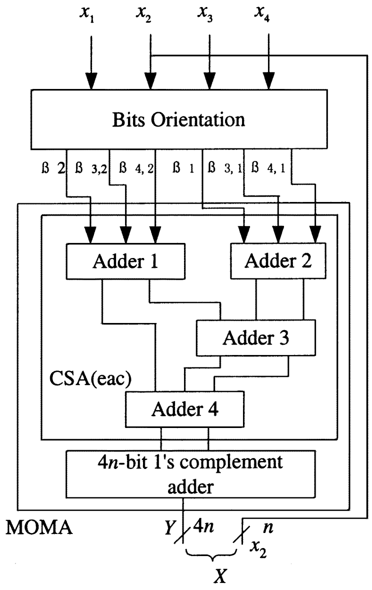

Fig. 1. Hardware realization of the proposed reverse converter.

modulo $2^{4n}-1$ adder, which can be implemented as efficiently as the algorithm of triple moduli set $\{2^n-1, 2^n, 2^n+1\}$. The complicated arithmetics in the algorithm of [18], [19] have been eliminated with the aid of *Property 1*. Similarly, efficient architecture as that used in the triple moduli set can be employed to realize our new reverse converter. With larger dynamic range and higher parallelism, more promising avenues for optimization at the CSA tree than that of the triple moduli set are envisaged.

Example 1: For $n = 3$, the moduli set is $\{7, 8, 9, 65\}$. Let $X = (3, 6, 7, 20)$. In binary string notation, $x_1 = (011)_2$, $x_2 = (110)_2$, $x_3 = (0111)_2$, $x_4 = (0010100)_2$. Based on (11)–(16), $\beta_1 = (11\ 011\ 011\ 011\ 0)_2 = 3510$, $\beta_2 = (001\ 111\,111\,111)_2 = 1023$, $\beta_{3,1} = (00\ 11\ 1000\ 11\ 10)_2 = 910$, $\beta_{3,2} = (0\ 0111\ 00\ 0111\ 0)_2 = 910$, $\beta_{4,1} = (0100\ 00\,000\ 001)_2 = 1025$, $\beta_{4,2} = (111\ 1\,101\,011\ 11)_2 = 4015$. From (17), $Y = |\beta_1 + \beta_2 + \beta_{3,1} + \beta_{3,2} + \beta_{4,1} + \beta_{4,2}|_{4095} = 3203$. From (9), $X = x_2 + 2^3 \times Y = 6 + 8 \times 3203 = 25\,630$. The causality is verified by $|25\,630|_7 = 3$, $|25\,630|_8 = 6$, $|25\,630|_9 = 7$ and $|25\,630|_{65} = 20$.

Fig. 1 shows the architecture of our reverse converter for the 4-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$. The bits orientation block generates $\beta_1, \beta_2, \beta_{3,1}, \beta_{3,2}, \beta_{4,1}, \beta_{4,2}$ by simply manipulating the routings of the bits from the input residue numbers of x_1, x_2, x_3 and x_4. The summation can be done by $(6, 2^{4n}-1)$ multioperand modular adder (MOMA) [6], which consists of a 4 n-bit 3-level CSA with EAC, and a 4 n-bit 1's complement adder [21]. It should be noted that as Y is weighted by 2^n, the addition of x_2 in (9) incurs no additional hardware and computation cost as it can be directly wired to the right of Y.

V. Performance Evaluation

In this section, we will estimate the hardware costs and evaluate the delay of the reverse converter. For ease of reference to relevant reverse converters, the standard practice of measuring complexity in terms of the number and delay of fundamental logic units like FA, standard logic gates, etc., is adopted for the reverse converters with generic n. The wire loads are normally neglected.

From Section IV, we know that β_1, β_2, $\beta_{3,1}$, $\beta_{3,2}$, $\beta_{4,1}$, and $\beta_{4,2}$ are all 4 n-bit wide, they can be obtained by hardwiring of the bits from the residue x_1, x_2, x_3, and x_4 and the supply (logic 1) and ground (logic 0). In the bits orientation block of Fig. 1, there are n inverters used for β_2, $n+1$ for $\beta_{3,1}$ and 2 $n+1$ for $\beta_{4,2}$. The total number of inverters is $4n+2$. The delay of this block is equal to $t_{\rm INV}$, which is the delay of an inverter.

It is also observed that there are strings of consecutive "1"s and "0"s embedded in the binary expressions of β_2, $\beta_{3,1}$, $\beta_{3,2}$, $\beta_{4,1}$, and $\beta_{4,2}$. Without annihilating the basic simplistic architecture of Fig. 1, a first cut simplification of the Carry-Save-Adders can be accomplished by manipulating those embedded constant strings. The fundamental idea is: a FA with a constant input "1" can be reduced to a pair of two-input XNOR and OR gates, a FA with a constant input "0" can be reduced to a pair of two-input XOR and AND gates, and a FA with an input "0" and an input "1" can be reduced to an inverter. One straightforward way to take advantage of the constant strings is to regroup each of β_2, $\beta_{3,2}$ and $\beta_{4,2}$ into five asymmetrical segments as follows:

$$\begin{aligned}
\beta_2 &= \underbrace{\overline{X}_{2,n-1}\cdots\overline{X}_{2,0}}_{n}\underbrace{11\cdots 11}_{3n}\\
&= \underbrace{\overline{X}_{2,n-1}\cdots\overline{X}_{2,0}}_{n}\underbrace{11}_{2}\underbrace{1\cdots 1}_{n-1}\underbrace{1\cdots 1}_{n+1}\underbrace{1\cdots 1}_{n-2}\\
\beta_{3,2} &= 0\underbrace{X_{3,n}\cdots X_{3,0}}_{n+1}\underbrace{0\cdots 0}_{n-1}\underbrace{X_{3,n}\cdots X_{3,0}}_{n+1}\underbrace{0\cdots 0}_{n-2}\\
&= \underbrace{0X_{3,n}\cdots X_{3,2}}_{n}\underbrace{X_{3,1}X_{3,0}}_{2}\underbrace{0\cdots 0}_{n-1}\underbrace{X_{3,n}\cdots X_{3,0}}_{n+1}\underbrace{0\cdots 0}_{n-2}\\
\beta_{4,2} &= \underbrace{1\cdots 1}_{n}\underbrace{\overline{X}_{4,2n}\overline{X}_{4,2n-1}}_{2}\underbrace{\overline{X}_{4,2n-2}\cdots\overline{X}_{4,n}}_{n-1}\\
&\quad\underbrace{\overline{X}_{4,n-1}\cdots\overline{X}_{4,0}1}_{n+1}\underbrace{1\cdots 1}_{n-2}.
\end{aligned}$$

If β_2, $\beta_{3,2}$ and $\beta_{4,2}$ are the chosen addends in the first level of the CSA tree, $n-1$ FAs can be eliminated from the rightmost (last) segment as they perform only the constant additions of "1" + "0" + "1" and their sum and carry outputs can be anticipated a priori without any computation. The $n-1$ "0"s in the generated sum and $n-1$ "1"s in the generated carry can be used to further reduce some FAs in Adder3 and Adder4 in the succeeding layer. In summing the fourth segments of β_2, $\beta_{3,2}$ and $\beta_{4,2}$ from the left, one input of each of these $n+1$ FAs is a "1," so these FAs can be reduced to $n+1$ pairs of two-input XNOR and OR gates. To add the third segments of these binary strings from the left, the $n-1$ FAs can be replaced by $n-1$ inverters as each adder has a constant "1" and a constant "0" inputs. For the first two segments from the left, the $n+2$ FAs can be substituted by $n+2$ pairs of XNOR and OR gates as each adder has one of its inputs equals "1." In summary, the number of FAs in Adder1 is 0, the number of XNOR and OR gates is 2 $n+3$ each, and the number of inverters is $n-1$.

Similarly, $\beta_{3,\,1}$ and $\beta_{4,\,1}$ can be segmented as follows:

$$\beta_{3,\,1} = \underbrace{\overline{X}_{3,\,1}\overline{X}_{3,\,0}}_{2}\underbrace{1\cdots1}_{n-1}\underbrace{\overline{X}_{3,\,n}\cdots\overline{X}_{3,\,0}}_{n+1}\underbrace{1\cdots1}_{n-2}$$
$$\underbrace{1\overline{X}_{3,\,n}\cdots\overline{X}_{3,\,2}}_{n}$$
$$\beta_{4,\,1} = \underbrace{X_{4,\,n}X_{4,\,n-1}}_{2}\underbrace{X_{4,\,n-2}X_{4,\,n-1}\cdots X_{4,\,0}}_{n-1}$$
$$\underbrace{0\cdots0}_{n+1}\underbrace{0\cdots0}_{n-2}\underbrace{X_{4,\,2n}X_{4,\,2n-1}\cdots X_{4,\,n+1}}_{n}.$$

Let $\beta_1, \beta_{3,\,1}$, and $\beta_{4,\,1}$ be added together in Adder2. The $n-1$ FAs for summing the fourth segments can be replaced by $n-1$ inverters due to the constant inputs of "0" and "1" to the adders. The $n+1$ FAs for adding the third segments are reduced into $n+1$ pairs of XOR and AND gates, because of the "0" input to each adder. The $n-1$ FAs for adding the second segments can be replaced by $n-1$ pairs of XNOR and OR gates, since one of every adder's inputs is a "1." Thus, the number of FAs in Adder2 is $n+2$, the number of XNOR and OR gates is $n-1$ each, the number of XOR and AND gates is $n+1$ each, and the number of inverters is $n-1$.

The number of FAs for the Adder3 and Adder4 in Fig. 1 is $4n-(n-2) = 3n+2$ each. As the $n-2$ sums and $n-2$ carries from Adder1 are all "0"s and all "1"s, respectively, the $n-2$ FAs are reduced to $n-2$ pairs of XOR/AND gates and XNOR/OR gates, accordingly.

As a result, the total number of FAs of the CSA with EAC is $7n+6$, the number of pairs of two-input XOR/AND gates is $2n-1$, the number of pairs of two-input XNOR/OR gates is $4n$, and the number of inverters is $2n-3$. The maximum delay of the CSA with EAC is $3t_{\text{FA}}$, where t_{FA} is the delay of a FA.

Overall, the hardware cost of our residue-to-binary converter for the proposed 4-moduli set is equivalent to $7n+6$ FAs, $2n-1$ pairs of two-input XOR/AND gates, $4n$ pairs of two-input XNOR/OR gates, $6n-1$ inverters, and one $4n$-bit carry propagate adder (CPA). It should be noted that further reduction in hardware complexity is possible if similarly weighted bits are swapped across β_i to optimize the use of constant inputs. The total delay of this reverse converter is $t_{\text{inv}} + 3t_{\text{FA}} + 2t_{\text{CPA}(4n)}$, if the method in [14] is adopted for the CPA, and this version of reverse converter is called the cost-effective (CE) version. The delay can be lowered to $t_{\text{inv}}+3t_{\text{FA}}+t_{\text{CPA}(2n)}+t_{\text{MUX}}+2t_{\text{NAND}}$ at the expense of increasing hardware cost when the method in [21] for CPA is adopted, which is called as the high-speed (HS) version of the converter.

It is obvious that the total delay of the reverse converter strongly depends on the delay of the CPA. If the fast ripple carry adder (RCA) proposed in [22] is used for the CPA, the delay of the $4n$-bit CPA will be $t_{\text{CPA}(4n)} \approx 4nt_{\text{NAND}}$. If a carry lookahead adder (CLA) is used for the CPA, then for $n \leq 16$, a two-level CLA is needed, and $t_{\text{CPA}(4n)} \approx 12$ unit delay; for $n > 16$, higher level CLA is needed, and each additional level of CLA will contribute 4-unit delay, where a 1-unit delay is the delay of a two-input NAND gate [23].

In [18], a residue-to-binary converter for moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$ is proposed. For this 4-moduli set, the total dynamic range is only $4n+1$ bits. Furthermore, its moduli are pairwise relatively prime only when $n = 2k+1$ where k is a positive integer. To represent some weighted binary number with the dynamic range just exceeding $4n{+}1$, the next value of n will have to be increased by 2, resulting in unnecessary wastage of the extended dynamic range and the associated computational cost. Excluding the delay of multiplexer, ROM lookup tables, and XOR gates, the delay of this converter is approximately $10n+13$ FAs delay. Even if the CPA of our proposed converter uses the slower traditional RCA [23], the total delay is estimated to be $(8n+3)t_{\text{FA}}$ and $(2n+3)t_{\text{FA}}$ for CE version and HS version, respectively, which is still faster than the converter of [18]. In comparing the area complexity of our proposed design, the size of the reduced FA cells must be taken into consideration. A reduced FA consists of either a pair of XOR/AND gates or a pair of XNOR/OR gates and a FA consists of two XORs and some NAND gates. Therefore, a reduced FA can be considered as compatible in area to half of the normal FA. The hardware cost of $2n-1$ pairs of two-input XOR/AND gates, $4n$ pairs of two-input XNOR/OR gates and $6n-1$ inverters is equivalent to $3n$ FAs, thus the area complexity of our new converter is $O(14n)$ for CE version and $O(18n)$ for HS version, which is more cost effective than the converter of [18], which is $O(n^2)$.

TABLE I
COMPARISONS OF HARDWARE REQUIREMENTS FOR REVERSE CONVERTERS OF 4-MODULI SETS

Converters	Hardware cells		Operand width	Amount	Total area
[19]	Binary Adder		$n+2$	1	$O(37n)$
	Binary Adder		$n+3$	1	
	Binary Adder		$2n+3$	2	
	Binary Adder		$3n$	1	
	Binary Adder		$3n+1$	1	
	Binary Subtractor		$n+3$	1	
	Binary Subtractor		$2n$	1	
	Binary Subtractor		$2n+1$	1	
	Binary Subtractor		$3n$	1	
	Binary Subtractor		$5n+1$	1	
	Modular Adder		$2n+3$	1	
	Modular Subtractor		$3n+1$	1	
[18]	N. A.		N. A.	N. A.	$O(n^2)$
Proposed	CE	CPA	$4n$	1	$O(14n)$
		CSA	$4n$	4	
	HS	CPA	$4n$	2	$O(18n)$
		CSA	$4n$	4	

N. A: Not available from [18]

A different 4-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ reverse converter has been proposed in [19] with a dynamic range of $4n$ bits. This time n has to be even, and it suffers from the same disadvantages as the moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$ in the choice of n for a fixed dynamic range. The total delay is $(14n+8)t_{\text{FA}}$, and the design consists of 13 adders/subtractors with bit width ranging from $n+1$ to $5n+1$ bits.

Table I summarizes the hardware requirements of the reverse converters for these 4-moduli sets. Only the number of adders and their operand width are considered for the converter of [19] and ours. This is because for these two converters, the total hardware costs are contributed predominantly by the adders. The simplification method of FAs is also adopted for evaluation of the area complexity of [19]. Furthermore, we assume that the

TABLE II
DELAY COMPARISONS OF THE REVERSE CONVERTERS FOR 4-MODULI SETS

Converters	Delay
[18]	$t_{MUX} + t_{LUT} + t_{XOR} + (10n + 13)*t_{FA}$
[19]	$(14n + 8)*t_{FA}$
Proposed – HS	$2t_{MUX} + t_{INV} + 2t_{NAND} + (2n + 3)*t_{FA}$
Proposed – CE	$t_{MUX} + t_{INV} + 2t_{NAND} + (8n + 3)*t_{FA}$

modular adder proposed in [24] will be used for the modular adder/subtractor in [19]. Due to heterogeneous operand width and varying types of adders and subtractors used in [19], it is conjectured that the scalability and modularity of [19] are inferior to ours. The reverse converter in [18] is implemented with ROM and modular adders, the area complexity of $O(n^2)$ is reported as the equivalent area occupied by FAs from the fabricated circuits.

Table II compares the latencies of these four-moduli set reverse converters. It should be noted that for a specified dynamic range, the value of n for our new converter is generally smaller than the n for the other converters.

VI. CONCLUSION

In the past decade, the triple moduli set $\{2^n - 1, 2^n, 2^n + 1\}$ has been the perpetual focus of study in simplifying the computationally intensive residue arithmetics for the reverse converter design of RNS. The revolutionary New CRT I have painted a completely different landscape for the RNS and strengthened the resurgence of interest for new moduli sets that have never been thought of before. In an earnest attempt to leverage on the strength of the new CRT, we have discovered a new 4-moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$. We show that this new moduli set possesses a number of interesting characteristics that make its reverse conversion algorithm under the New CRT I amenable to efficient VLSI implementation. The new reverse converter completely eliminates the need for modulo multiplication and allows further optimization opportunity in a simple MOMA realization. With an area complexity of $O(18n)$ and a delay of approximately $2\ n + 3$ FAs for HS version, and with an area complexity of $O(14n)$ and a delay of approximately $8n + 3$ FAs for CE version, they are more efficient than the reverse converter for the celebrated triple moduli set $\{2^n - 1, 2^n, 2^n + 1\}$ for applications requiring a large dynamic range and high parallelism. The area time complexity analysis also indicates that our reverse converters are more efficient than the converters for the 4-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$ and $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$, both in hardware area and computation delay.

REFERENCES

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

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

[3] S. Perumal and R. E. Siferd, "Pipelined 50 MHz CMOS ASIC for 32-bit binary to residue conversion and residue-to-binary conversion," in *Proc. 7th Annual IEEE Int. ASIC Conf. And Exhibit*, Rochester, NY, Sept. 1994, pp. 454–457.

[4] T. Srikanthan, M. Bhardwaj, and C. T. Clarke, "Area-time-efficient VLSI residue-to-binary converters," *Proc. Inst. Elect. Eng. Comput. Digit. Tech.*, vol. 145, no. 3, pp. 229–235, 1998.

[5] S. R. Barraclough *et al.*, "The design and implementation of the IMS A110 image and signal processor," in *Proc. IEEE Custom Integr. Circuits Conf.*, 1989, pp. 24.5.1–24.5.4.

[6] S. J. Piestrak, "Design of residue generators and multioperand modular adders using carry-save adders," *IEEE Trans. Comput.*, vol. 43, pp. 68–77, Jan. 1994.

[7] G. Alia and E. Martinelli, "A VLSI algorithm for direct and reverse conversion from weighted binary number system to residue number system," *IEEE Trans. Circuits Syst.*, vol. CAS-31, pp. 1033–1039, Dec. 1984.

[8] R. M. Capocelli and R. Giancarlo, "Efficient VLSI networks for converting an integer from binary system to residue number system and vice versa," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1425–1430, Nov. 1988.

[9] K. M. Elleithy and M. A. Bayoumi, "Fast and flexible architectures for RNS arithmetic decoding," *IEEE Trans. Circuits Syst. II*, vol. 39, pp. 226–235, Apr. 1992.

[10] F. Barsi and M. C. Pinotti, "A fully parallel algorithm for residue-to-binary conversion," in *Proc. Information Lett.*, vol. 50, 1994, pp. 1–8.

[11] C. H. Huang, "A fully parallel mixed-radix conversion algorithm for residue number applications," *IEEE Trans. Comput.*, vol. 32, pp. 398–402, Apr. 1983.

[12] H. M. Yassine and W. R Moore, "Improved mixed-radix conversion for residue number system architectures," in *Proc. Inst. Elect. Eng. -G*, vol. 138, 1991, pp. 120–124.

[13] S. Andraos and H. Ahmad, "A new efficient memoryless residue-to-binary converter," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1441–1444, Nov. 1988.

[14] S. J. Piestrak, "A high-speed realization of a residue-to-binary number system converter," *IEEE Trans. Circuits Syst. II*, vol. 42, pp. 661–663, Oct. 1995.

[15] Y. Wang, "New Chinese remainder theorems," in *Proc. 32th Asilomar Conf. Signals, Systems, Computers*, vol. 1, 1998, pp. 165–171.

[16] ——, "Residue-to-binary converters based on new Chinese Remainder Theorems," *IEEE Trans. Circuits Syst. II*, vol. 47, pp. 197–205, Mar. 2000.

[17] Y. Wang, X. Song, M. Aboulhamid, and H. Shen, "Adder based residue-to-binary number converters for (2^nt 1, 2^n, $2^n + 1$)," *IEEE Trans. Signal Processing*, vol. 50, pp. 1772–1779, July 2002.

[18] M. Bhardwaj, T. Srikanthan, and C. T. Clarke, "A reverse converter for the 4-moduli superset $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$," in *Proc. 14th IEEE Symp. Computer Arithmetic*, Adelaide, Australia, Apr. 1999, pp. 168–175.

[19] A. P. Vinod and A. B. Premkumar, "A memoryless reverse converter for the 4-moduli superset $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$," *J. Circuits, Syst., Comput.*, vol. 10, no. 1&2, pp. 85–99, 2000.

[20] N. Robbins, *Beginning Number Theory*. Dubuque, IA: William. C. Brown, 1993.

[21] M. Bhardwaj, A. B. Premkumar, and T. Srikanthan, "Breaking the 2 n-bit carry propagation barrier in residue-to-binary conversion for the $\{2^n - 1, 2^n, 2^n + 1\}$ moduli set," *IEEE Trans. Circuits Syst. I*, vol. 45, pp. 998–1002, Sept. 1998.

[22] I. S. Reed *et al.*, "VLSI implementation of GSC architecture with a new ripple carry adder," in *Proc. ICCD'88*, 1988, pp. 520–523.

[23] K. Hwang, *Computer Arithmetic: Principle, Architecture and Design*. New York: Wiley, 1979.

[24] S. J. Piestrak, "Design of high-speed residue-to-binary number system converter based on Chinese Remainder Theorem," in *Proc. 94th Int. Conf. Computer Design*, Oct. 1994, pp. 508–511.

Bin Cao received the B.Eng. degree from the Wuhan Institute of Technology, Wuhan, China, in 1991, the M.Eng. degree from Zhejiang University, Hangzhou, China, in 1996. He is currently working toward the Ph.D. degree at Nanyang Technological University, Singapore.

From July 1996 to October 2001, he was with Eastern Communications Company, Hangzhou, China. His research interests include computer arithmetic and VLSI digital signal processing.

Chip-Hong Chang (S'93–M'98–SM'03) received the B.Eng. (Hons.) degree in electrical engineering from the National University of Singapore, Singapore, in 1989, and the M.Eng. and Ph.D. degrees in electrical and electronic engineering from Nanyang Technological University, Singapore, in 1993 and 1998, respectively.

His industrial experience includes Component Engineer of General Motors, Singapore and Technical Consultant of Flextech Electronics Pvt. Ltd, Singapore. He joined the Design Centre of Nanyang Polytechnic University, Singapore, as a Lecturer in Electronics, in 1993. Since 1999, he has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, where he is currently an Assistant Professor. He holds concurrent appointments at the university as the Deputy Director of the Centre for High Performance Embedded Systems, and the Program Director of VLSI Design and Embedded Systems Research Group of the Centre for Integrated Circuits and Systems. His current research interests include low-power arithmetic circuits, design automation and synthesis, and algorithms and architectures for digital image processing. He has published about 70 refereed international journal and conference papers.

Thambipillai Srikanthan (SM'93) received the B.Sc. (Hons.) degree in computer and control systems and the Ph.D. degree in system modeling and information systems engineering from Coventry University, Coventry, U.K.

He has been with the Nanyang Technological University (NTU), Singapore, since 1991, where he holds a joint appointment as Associate Professor and Director of the Centre for High Performance Embedded Systems. He is the Founder and Director of the Centre for High Performance Embedded Systems, which is now a University level research center at NTU. His research interests include system integration methodologies, architectural translations of compute intensive algorithms, high-speed techniques for image processing and dynamic routing. He has published more than 100 technical papers and has served a number of administrative roles during his academic career.

Dr. Srikanthan is a corporate member of the Institution of Electrical Engineers.