An Efficient Residue to Weighted Converter for a New Residue Number System

Alexander Skavantzos
Department of Electrical and Computer Engineering
Louisiana State University, Baton Rouge, LA 70803
Phone: (504) 388-5240; Fax: (504) 388-5200
e-mail: alex@ee.lsu.edu

Abstract

The Residue Number System (RNS) is an integer system appropriate for implementing fast digital signal processors since it can support parallel, carry-free, high-speed arithmetic. In this paper a new RNS system and an efficient implementation of its residue-to-weighted converter are presented. The new RNS is a balanced 5-moduli system appropriate for large dynamic ranges. The new residue-to-binary converter is very fast and hardware-efficient and is based on a 1's complement multioperand adder adding operands of size only 80% of the size of the system's dynamic range.

1. Introduction and Background

The Residue Number System (RNS) [1] is an integer system capable of supporting parallel, carry-free, high-speed arithmetic. The system also offers some useful properties for error detection, error correction and fault tolerance in digital systems. Important areas of application of the RNS include:

1. Digital Signal Processing (DSP) intensive computations such as digital filtering, convolutions, correlations and DFT and FFT computations [2]-[13].
2. Direct Digital Frequency Synthesis [14].

Recent work in RNS arithmetic has resulted in the development of the Quadratic Residue Number System (QRNS) [15]-[16], [7], [17]-[18] the Quadratic Like Residue Number System (QLRNS) [19], the Modified Quadratic Residue Number System (MQRNS) [20] and the generalization of the above Quadratic Residue Systems, the Polynomial Residue Number System (PRNS) [21]-[22], [9]-[10]. All these systems can support complex DSP operations using minimum computational complexity and maximum parallelism.

The basis for an RNS is a set of relatively prime integers

$$S = \{m_1, m_2, \ldots, m_L\}, \text{ where } (m_i, m_j) = 1 \text{ for } i \neq j \qquad (1)$$

with (m_i, m_j) indicating the greatest common divisor of m_i and m_j. The set S is called the moduli set, while the dynamic range of the system is defined by the product M of all the moduli m_i in the set S. Any integer X belonging to $Z_M = \{0,1,2,\ldots,M-1\}$ has a unique RNS representation given by

$$X \xrightarrow{RNS} (X_1, X_2, \ldots, X_L) \qquad (2)$$

where

$$X_i = \langle X \rangle_{m_i}, \; i = 1,2,\ldots,L \qquad (3)$$

while $\langle x \rangle_m$ denotes the operation x mod m. If the integers X and Y have RNS representations $(X_1,\ldots,X_L)$ and $(Y_1,\ldots,Y_L)$ respectively, then the RNS representation of $W = X \otimes Y$ (where $\otimes$ denotes addition, subtraction or multiplication) is given by

$$W \xrightarrow{RNS} (W_1,\ldots,W_L); W_i = \langle X_i \otimes Y_i \rangle_{m_i}, i = 1,\ldots,L \qquad (4)$$

Equation (4) demonstrates the parallel, carry-free nature of the RNS. It must be mentioned that in order to ensure fast internal RNS processing, the moduli $m_1, m_2, \ldots m_L$ should be as small as possible. The reconstruction of X from its residues $(X_1, X_2, \ldots, X_L)$ is based on the Chinese Remainder Theorem (CRT) [1] shown by equation (5)

$$X = \left\langle \sum_{i=1}^{L} \langle X_i N_i \rangle_{m_i} M_i \right\rangle_M \qquad (5)$$

where

$$M = \prod_{i=1}^{L} m_i \qquad (6)$$

$$M_i = \frac{M}{m_i}; N_i = \langle M_i^{-1} \rangle_{m_i}, i = 1,2,\ldots,L \qquad (7)$$

0-8186-8409-7/98 $10.00 © 1998 IEEE

The notation $\left\langle M_i^{-1} \right\rangle_{m_i}$ in equation (7) denotes the multiplicative inverse of M_i modulo m_i. Another way for converting the RNS representation $(X_1, X_2, \ldots, X_L)$ into its weighted form X is by using the Mixed Radix Conversion (MRC) formula [1] shown by equation (8)

$$X = X_1' + m_1 X_2' + m_1 m_2 X_3' + \ldots + m_1 m_2 \ldots m_{L-1} X_L' \qquad (8)$$

where $X_1', X_2', \ldots, X_L'$ are the mixed radix digits of X.

In this paper a very efficient new residue-to-weighted conversion technique is proposed. The technique is based on combining the CRT and MRC techniques and relies on a final adder of size smaller than the size of the system's dynamic range. Section 2 offers the new decoding technique and the class of RNS systems which are appropriate for the new technique. A new 5-moduli RNS system and the efficient implementation of its residue-to-weighted converter are presented in section 3. Finally, section 4 offers conclusions.

2. RNS Systems with Efficient Residue to Weighted Conversion

Consider an L-moduli RNS system based on the moduli set S of (1) and consider converting the residue form $(X_1, X_2, \ldots, X_L)$ into its weighted form X by using the Chinese Remainder Theorem (CRT) of (5). An implementation of the CRT equation (5) can be based on a multioperand adder mod M (M is the system's dynamic range given by (6)). Such a mod M multioperand adder can be efficiently implemented by a mod M Carry Save Adder (CSA) tree and a mod M Carry Propagate Adder (CPA). Let N be the size of the dynamic range M in terms of number of bits. Then

$$N = \lceil \log_2 M \rceil \qquad (9)$$

The following observations are in place:

1. The size N directly affects the speed and cost of the mod M CRT multioperand adder of (5). The larger the size N is, the higher the cost of the mod M CSA tree becomes. Also, the larger the size N is, the higher the cost and the propagation delay for the final mod M CPA becomes.
2. The form of the number M (the dynamic range) affects the speed and hardware complexity of the mod M CRT multioperand adder.

A new class of RNS systems appropriate for very efficient residue-to-weighted conversion is now presented. The RNS-to-weighted converters for these new systems rely on a hardware-efficient multioperand modulo adder of size smaller than the system's dynamic range.

Consider an RNS system based on the moduli set S of (1) where one of the moduli (say the modulus m_1) is a power of two or

$$m_1 = 2^i \qquad (10)$$

The conversion of the RNS representation $(X_1, X_2, \ldots, X_L)$ into its weighted form X will now take place by using a combination of the CRT and MRC techniques as follows:

Combine the channels mod m_2, mod m_3, . . . , mod m_L using the CRT approach. The set used for this CRT is

$$S^* = \{m_2, m_3, \ldots, m_L\} \qquad (11)$$

and the performed CRT computation is

$$X_2^* = \left\langle \sum_{i=2}^{L} \left\langle X_i N_i^* \right\rangle_{m_i} M_i^* \right\rangle_{M^*} \qquad (12)$$

where

$$M^* = \prod_{i=2}^{L} m_i \qquad (13)$$

$$M_i^* = \frac{M^*}{m_i}, N_i^* = \left\langle \left(M_i^* \right)^{-1} \right\rangle_{m_i}, i = 2, 3, \ldots, L \qquad (14)$$

Apply now the MRC formula on channels mod m_1 and mod M^* (see (10), (13) for m_1, M^*) to get

$$X = X_1' + m_1 X_2'^* \qquad (15)$$

where

$$X_1' = X_1 \qquad (16)$$

$$X_2'^* = \left\langle m_1^{-1} \left(X_2^* - X_1 \right) \right\rangle_{M^*} \qquad (17)$$

Combining (17) and (12) yields

$$X_2'^* = \left\langle m_1^{-1} \left[\left(\sum_{i=2}^{L} \left\langle X_i N_i^* \right\rangle_{m_i} M_i^* \right) - X_1 \right] \right\rangle_{M^*} \qquad (18)$$

Equation (18) dictates that $X_2'^*$ can be computed by a mod M^* multioperand adder of size N^* bits where N^* is

$$N^* = \lceil \log_2 M^* \rceil = \left\lceil \log_2 \frac{M}{2^i} \right\rceil = N - i \qquad (19)$$

According to (15), (16) the desired X can be computed by

$$X = X_1 + m_1 X_2'^* \qquad (20)$$

Since $m_1 = 2^i$ (see (10)) and the residue X_1 is an i-bit number, $\left(X_1 = \langle X \rangle_{m_1}\right)$, no computational hardware is needed to compute X according to equation (20). The desired X is just the result of concatenating $X_2'^*$ and X_1 or

$$X = X_2'^*, X_1 \qquad (21)$$

where comma (,) denotes concatenation. Thus, using the novel RNS decoding technique of (18) and (21) (which is based on combining CRT and MRC), the RNS-to-weighted conversion can rely on a multioperand modulo adder of size smaller than the size of the system's dynamic range. This is possible if one of the L moduli in the system is of form 2^i. In this case, the size of the multioperand adder will be by i bits less than the size of the system's dynamic range (see (19)).

Our next concern is that the mod M^* multioperand adder of equation (18) be as fast and hardware-efficient as possible. This will of course depend on the form of the number M^* (see (13) for M^*). The best form of M^* is $M^* = 2^a$. This is mathematically impossible, however, due to the fact that $M^* = 2^a$ can not be factored into pairwise relatively prime integers $m_2, m_3, \ldots, m_L$. The second most attractive M^* is the form $M^* = 2^a - 1$ which is the choice considered by this paper.

3. A New RNS and its Converter Design

As seen in section 2, if a multimoduli RNS system relies on a moduli set with one modulus being of form 2^i and the product of the remaining moduli being of form $M^* = 2^a - 1$, then the RNS-to-weighted conversion can rely on an efficient adder of size smaller than the size of the system's dynamic range. This is possible due to the presented novel RNS decoding technique which is based on combining the CRT and MRC techniques. The simplest such RNS system is the 2-moduli system with $m_1 = 2^n$ and $M^* = m_2 = 2^n - 1$. Another system of this category is the popular 3-moduli system with $m_1 = 2^n$ and $M^* = m_2 m_3 = 2^{2n} - 1$ which implies $m_2 = 2^n - 1$ and $m_3 = 2^n + 1$ [23]-[27]. Both the above mentioned systems rely on simple RNS-to-weighted conversion but are not appropriate for large dynamic ranges. This is due to the fact that in case of large dynamic ranges, large values of n are required resulting in apparent performance degradation of the system.

For large dynamic ranges, RNS systems with more than two or three moduli must be considered. A new 5-moduli RNS appropriate for efficient residue-to-weighted conversion is now presented. The new system is based on the set S_1

$$S_1 = \{m_1, m_2, m_3, m_4, m_5\} = \{2^{n+1}, 2^n - 1, 2^n + 1, 2^n + 2^{\frac{n+1}{2}} + 1, 2^n - 2^{\frac{n+1}{2}} + 1\}, n \text{ is odd integer} \qquad (22)$$

Here

$$m_1 = 2^{n+1} \qquad (23)$$

$$M^* = \prod_{i=2}^{5} m_i = 2^{4n} - 1 \qquad (24)$$

The moduli $m_1, m_2, \ldots, m_5$ of the set S_1 are pairwise relatively prime while the achieved dynamic range (in number of bits) is $DRs_1 = 5n + 1$ bits. Also, the set S_1 implies balanced arithmetic since the sizes of the $\bmod m_1, \bmod m_2, \ldots, \bmod m_5$ processors are $n+1, n, n, n+1$ and n bits respectively.

Due to the fact that m_1 and M^* are forms dictated by (23)-(24), the new RNS of set S_1 (eq. (22)) can be decoded by using the novel technique of (18) and (21). Let the RNS representation of X be $(X_1, X_2, \ldots, X_5)$. Then (18) becomes

$$X_2'^* = \langle m_1^{-1}(AM_2^* + BM_3^* + CM_4^* + DM_5^* - X_1) \rangle_{M^*} \qquad (25)$$

where

$$M^* = 2^{4n} - 1 \qquad (26)$$

$$m_1 = 2^{n+1} \qquad (27)$$

$$\langle m_1^{-1} \rangle_{M^*} = 2^{3n-1} \qquad (28)$$

$$A = \langle X_2 N_2^* \rangle_{m_2} \qquad (29)$$

$$B = \langle X_3 N_3^* \rangle_{m_3} \qquad (30)$$

$$C = \langle X_4 N_4^* \rangle_{m_4} \qquad (31)$$

$$D = \langle X_5 N_5^* \rangle_{m_5} \qquad (32)$$

$$M_2^* = (2^n + 1)(2^{2n} + 1) \qquad (33)$$

$$M_3^* = (2^n - 1)(2^{2n} + 1) \quad (34)$$

$$M_4^* = (2^{2n} - 1)(2^n - 2^{\frac{n+1}{2}} + 1) \quad (35)$$

$$M_5^* = (2^{2n} - 1)(2^n + 2^{\frac{n+1}{2}} + 1) \quad (36)$$

where m_2, m_3, m_4, m_5 are given by (22) while $N_2^*, N_3^*, N_4^*, N_5^*$ are given by (14). It must be mentioned that the computations of A, B, C, and D (equations (29) - (32)) are performed by the existing $\bmod m_i$ $(i = 2,...,5)$ multipliers which belong to the RNS processing hardware. Since A, B, C, D, and X_1 belong to the rings of integers $\bmod(2^n - 1)$, $\bmod(2^n + 1)$, $\bmod(2^n + 2^{\frac{n+1}{2}} + 1)$, $\bmod(2^n - 2^{\frac{n+1}{2}} + 1)$ and $\bmod(2^{n+1})$, respectively, then these numbers are of lengths $n, n+1, n+1, n$ and $n+1$ bits respectively. Let the binary (bit-level) representations of A, B, C, D, and X_1 be

$$A = (a_{n-1}a_{n-2}....a_1a_0) \quad (37)$$

$$B = (b_nb_{n-1}b_{n-2}....b_1b_0) \quad (38)$$

$$C = (c_nc_{n-1}c_{n-2}....c_1c_0) \quad (39)$$

$$D = (d_{n-1}d_{n-2}....d_1d_0) \quad (40)$$

$$X_1 = (x_nx_{n-1}x_{n-2}....x_1x_0) \quad (41)$$

Combining equations (26), (28), (33) - (41) together with (25) and using simple properties of arithmetic $\bmod(2^a - 1)$ results in

$$X_2'^* = \langle \sum_{i=1}^{11} Z_i \rangle_{2^{4n}-1} \quad (42)$$

where $Z_1, Z_2, ..., Z_{11}$ are the following $4n - bit$ vectors

$$Z_1 = (a_0a_{n-1}a_{n-2}...a_1)^4 \quad (43)$$

$$Z_2 = b_0(0)^{n-1}b_nb_{n-1}...b_1b_0(0)^{n-1}b_nb_{n-1}...b_2b_1 \quad (44)$$

$$Z_3 = \overline{b}_n\overline{b}_{n-1}...\overline{b}_1\overline{b}_0(1)^{n-1}\overline{b}_n\overline{b}_{n-1}...\overline{b}_1\overline{b}_0(1)^{n-1} \quad (45)$$

$$Z_4 = c_{\frac{n-1}{2}}....c_0(0)^{\frac{n-1}{2}}c_n(c_{n-1}...c_1c_0)^2 \quad (46)$$
$$(0)^{\frac{n-3}{2}}c_n...c_{\frac{n+1}{2}}$$

$$Z_5 = \overline{c_0}\,\overline{c_{n-1}}...\overline{c_1}\,\overline{c_0}(1)^{\frac{n-3}{2}}\overline{c_n}...\overline{c_1}\,\overline{c_0} \quad (47)$$
$$(1)^{\frac{n-1}{2}}\overline{c_n}...\overline{c_2}\,\overline{c_1}$$

$$Z_6 = 1(\overline{c_n})^{2n}(1)^{2n-1} \quad (48)$$

$$Z_7 = (0)^{n+1}(d_{n-1}d_{n-2}...d_1d_0)^2(0)^{n-1} \quad (49)$$

$$Z_8 = (0)^{\frac{3n+1}{2}}d_{n-1}d_{n-2}...d_1d_0(0)^{\frac{3n-1}{2}} \quad (50)$$

$$Z_9 = \overline{d_0}\,\overline{d_{n-1}}...\overline{d_1}\,\overline{d_0}(1)^{2n}\overline{d_{n-1}}...\overline{d_2}\,\overline{d_1} \quad (51)$$

$$Z_{10} = \overline{d}_{\frac{n-1}{2}}...\overline{d_1}\,\overline{d_0}(1)^{3n}\overline{d_{n-1}}...\overline{d}_{\frac{n+1}{2}} \quad (52)$$

$$Z_{11} = \overline{x_n}\,\overline{x_{n-1}}...\,\overline{x_1}\,\overline{x_0}(1)^{3n-1} \quad (53)$$

In the equations (43) - (53) the notation $(0)^n$ indicates a string of n zeros, $(1)^n$ indicates a string of n ones while $(abcd)^k$ indicates a 4k-bit vector where the 4-bit string abcd is repeated k times. For example $(1)^3(ab)^2(0)^4$ means 111abab0000. In order to improve the readability of the paper, derivations for the expressions of $Z_1, Z_2, ...Z_{11}$ (equations (43) - (53)) are not provided here.

Figure 1 shows an efficient implementation of the new RNS-to-binary converter for the new 5-moduli system of (22). The converter implements equation (42) using the carry save adder (CSA) approach. The converter consists of a mod $\left(2^{4n} - 1\right)$ CSA tree and a mod $\left(2^{4n} - 1\right)$ carry propagate adder (CPA). Each mod $\left(2^{4n} - 1\right)$ CSA in the tree consists of $4n$ full adders (FAs). The outputs of each mod $\left(2^{4n} - 1\right)$ CSA are the $4n$-bit summation vector S_i and the $4n$-bit vector C_i, where C_i is the left-rotated by one bit carry-out vector (end-around carry). The presence of zeros and ones in the vectors of (43) - (53) implies that some of the full adders (FAs) in the CSAs can be replaced by simpler gates (AND, OR, XOR etc.) costing less and having smaller propagation delay than a FA. The repetition of certain

binary strings in the vectors of (43) - (53) implies that some CSAs will rely on fewer than $4n$ FAs or simplified FAs. The mod $\left(2^{4n}-1\right)$ CPA is a 1's complement carry propagate adder of size $4n$ bits. An area-time efficient design of such an adder is provided by reference [28]. The result of the 5-moduli RNS-to-weighted conversion is the concatenation of the $4n$-bit vector $X_2'^{*}$ and the $(n+1)$-bit residue X_1 (see figure 1 and equation (21)). The cost and delay for the new RNS-to-weighted converter are

$$\text{Converter-cost} < 36nC_1 + C_2 \qquad (54)$$

$$\text{Converter-delay} = 5D_1 + D_2 \qquad (55)$$

where C_1 and D_1 are the cost and delay for a full adder (FA) while C_2 and D_2 are the cost and delay for a mod $\left(2^{4n}-1\right)$ carry propagate adder (CPA). The parameter n is the number of bits per moduli channel; (the system has a dynamic range of $5n+1$ bits).

4. Conclusion

In this paper a class of multimoduli RNS systems appropriate for very efficient residue-to-weighted binary conversion has been presented. Specific emphasis was placed on a new balanced 5-moduli such RNS system and its converter design. The presented L-moduli RNS class relies on one modulus being of the form $m_1 = 2^i$ and the product of the remaining L-1 moduli being of form $M^* = 2^a - 1$. The proposed residue-to-weighted conversion technique is based on combining the Chinese Remainder Theorem (CRT) and the Mixed Radix Conversion (MRC) techniques. The resulting new RNS-to-weighted converters are very fast and hardware-efficient due to the following reasons:

1. The new converters rely on a multioperand adder of size smaller than the size of the system's dynamic range.
2. The multioperand adder is a mod $\left(2^a-1\right)$ adder.

A carry save adder (CSA) based implementation of a mod $\left(2^a-1\right)$ multioperand adder is faster and more hardware-efficient than the implementation of a mod M adder with $M \neq 2^a - 1$.

References

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

[2] W.K. Jenkins and B.J. Leon, "The use of residue number systems in the design of finite impulse response digital filters," *IEEE Transactions on Circuits and Systems*, vol. CAS-24, pp. 191-201, April 1977.

[3] M.A. Soderstrand, "A high-speed low-cost recursive digital filter using residue number arithmetic," *Proceedings of IEEE*, vol. 65, pp. 1065-1067, July 1977.

[4] W.K. Jenkins, "Recent advances in residue number techniques for recursive digital filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-27, pp. 19-30, February 1979.

[5] H.K. Nagpal, G.A. Jullien and W.C. Miller, "Processor architectures for two-dimensional convolvers using a single multiplexed computational element with finite field arithmetic," *IEEE Transactions on Computers*, vol. C-32, pp. 989-1000, November 1983.

[6] M.A. Soderstrand and B.Sinha, "A pipelined recursive residue number system digital filter," *IEEE Transactions on Circuits and Systems*, vol. CAS-31, pp. 415-417, April 1984.

[7] F.J. Taylor, G. Papadourakis, A. Skavantzos and A. Stouraitis, "A radix-4 FFT using complex RNS arithmetic," *IEEE Transactions on Computers*, vol. C-34, pp. 573-576, June 1985.

[8] A. Skavantzos, "Using quadratic residue arithmetic for computing skew cyclic convolutions," *Electronics Letters*, vol. 27, no. 23, pp. 2140-2141, November 1991.

[9] A. Skavantzos and N. Mitash, "Implementation issues of 2-dimensional polynomial multipliers for signal processing using residue arithmetic," *IEE Proceedings-E*, vol. 140, no. 1, pp. 45-53, January 1993.

[10] A. Skavantzos and T. Stouraitis, "Polynomial residue complex signal processing," *IEEE Transactions on Circuits and Systems-II*, vol. 40, no. 5, pp. 342-344, May 1993.

[11] C.-L. Wang, "New bit serial VLSI implementation of RNS FIR digital filters," *IEEE Transactions on*

Circuits and Systems-II, vol. 41, no. 11, pp. 768-772, November 1994.

[12] J.C. Smith and F.J. Taylor, "A fault-tolerant GEQRNS processing element for linear systolic array DSP applications," *IEEE Transactions on Computers*, vol. 44, no. 9, pp. 1121-1130, September 1995.

[13] M.A. Bayoumi, "A high speed VLSI complex digital signal processor based on quadratic residue number system," *VLSI Signal Processing II*, pp. 200-211, IEEE Press, 1986.

[14] W.A. Chren, "RNS-based enhancements for direct digital frequency synthesis," *IEEE Transactions on Circuits and Systems-II*, vol. 42, no. 8, pp. 516-524, August 1995.

[15] J.V. Krogmeier and W.K. Jenkins, "Error Detection and Correction in Quadratic Residue Number System," *Proceedings of the 26th Midwest Symposium on Circuits and Systems*, (Puebla, MX), pp. 408-411, August 1983.

[16] S.H. Leung, "Application of Residue Number systems to Complex Digital Filters," *Proceedings of Fifteenth Asilomar Conference on Circuits, Systems, and Computers*, (Pacific Grove, CA), pp. 70-74, November 1981.

[17] M. Abdallah and A. Skavantzos, "New multi-moduli residue and quadratic residue systems for large dynamic ranges," *Proceedings of 29th Asilomar Conference on Signals, Systems, and Computers*, (Pacific Grove, CA, October 1995), pp. 961-965.

[18] M. Abdallah and A. Skavantzos, "*On the binary quadratic residue system with non coprime moduli,*" IEEE Transactions on Signal Processing, vol. 45, no. 8, pp. 2085-2091, August 1997.

[19] M.A. Soderstrand and G.D. Poe, "Applications of Quadratic Like Complex Residue Number System Arithmetic to Ultrasonics," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, (San Diego, CA), pp. 28A.5.1-28A.5.4, March 1984.

[20] R. Krishnan, G.A. Jullien and W.C. Miller, "The Modified Quadratic Residue Number System (MQRNS) for Complex High-Speed Signal Processing," *IEEE Trans. on Circuits and Systems*, Vol. CAS-33, No. 3, pp. 325-327, March 1986.

[21] A. Skavantzos and F.J. Taylor, "On the Polynomial Residue Number System," *IEEE Transactions on Signal Processing*, Vol. 39, No. 2, pp. 376-382, February 1991.

[22] T. Stouraitis and A. Skavantzos, "*Multiplication of complex numbers encoded as polynomials,*" Journal of VLSI Signal Processing, vol. 3, no. 4, pp. 319-328, October 1991.

[23] F.J. Taylor and A.S. Ramnarayanan, "An efficient residue-to-decimal converter", *IEEE Transactions on Circuits and Systems*, vol. CAS-28, pp. 1164-1169, December 1981.

[24] P. Bernardson, "Fast memoryless, over 64 bits, residue-to-binary converter", *IEEE Transactions on Circuits and Systems*, vol. CAS-32, pp. 298-300, March 1985.

[25] K.M. Ibrahim and S.N. Saloum, "An efficient residue to binary converter design," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 9, pp. 1156-1158, September 1988.

[26] S. Andraos and H. Ahmad, "A new efficient memoryless residue to binary converter," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 11, pp. 1441-1444, November 1988.

[27] S.J. Piestrak, "A high-speed realization of a residue to binary number system converter," *IEEE Transactions on Circuits and Systems-II*, vol. 42, no. 10, pp. 661-663, October 1995.

[28] C.Efstathiou, D. Nikolos and J. Kalamatianos, "Area-time efficient modulo $2^n - 1$ adder design," *IEEE Trans. on Circuits and Systems - II*, vol. 41, no. 7, pp. 463-467, July 1994.

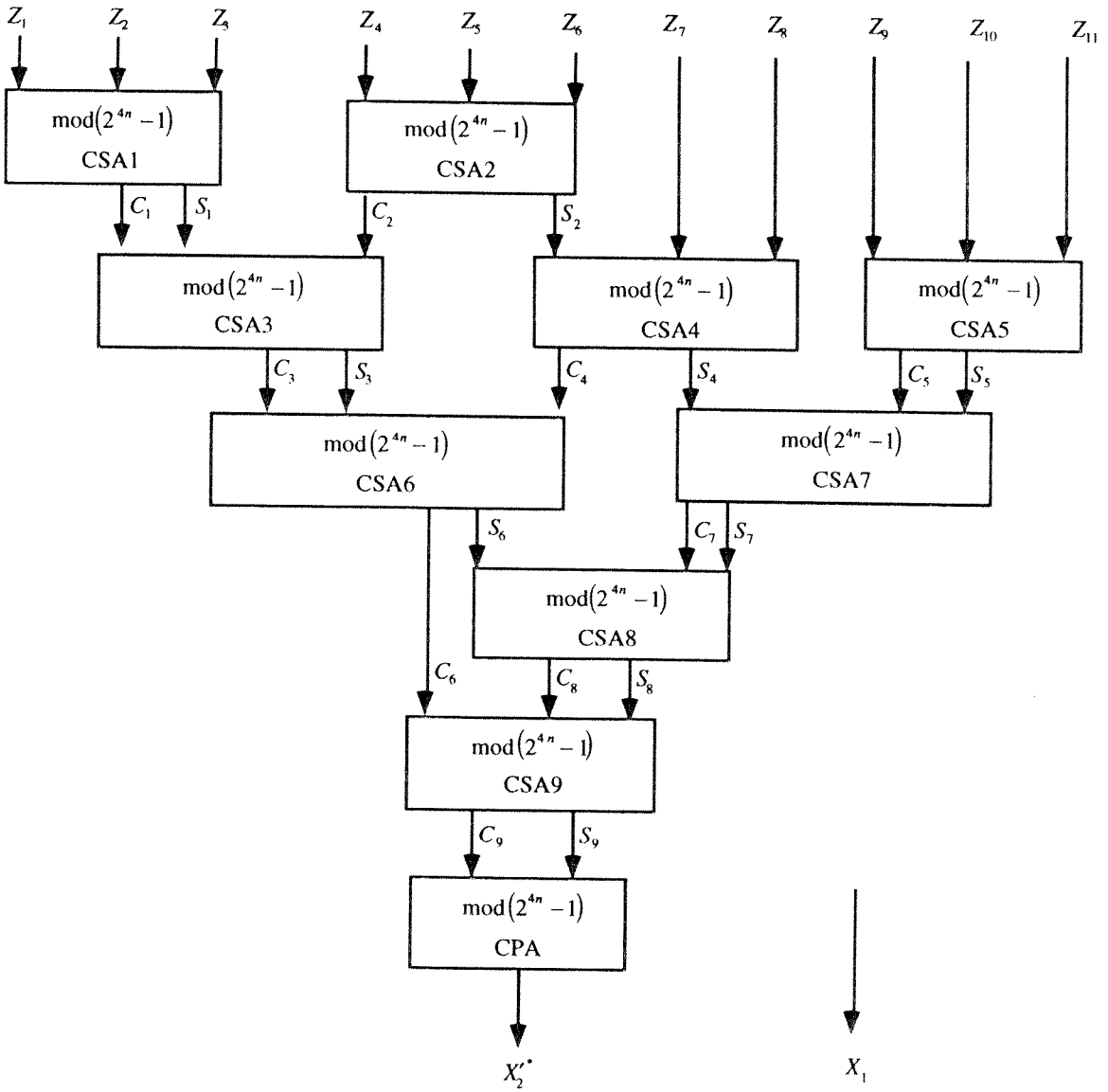

Figure 1: An efficient implementation of the new RNS-to-binary converter