Research Journal of Applied Sciences, Engineering and Technology 6(11): 2027-2031, 2013
DOI:10.19026/rjaset.6.3819
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: November 24, 2012 Accepted: January 19, 2013 Published: July 25, 2013

Research Article

Hybrid RNS-to-Binary Converter for the Moduli Set {2^{2n}, 2^n-1, 2^n+1, 2^{n+1}-1}

[1]Somayyeh Jafarali Jassbi and [2]Amir Sabbagh Molahosseini
[1]Faculty Member of Computer Engineering Department, Science and Research Branch,
Islamic Azad University, Tehran, Iran
[2]Department of Computer Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran

Abstract: The four-moduli Residue Number System (RNS) sets such as {2^n-1, 2^n, 2^n+1, 2^{n+1}-1} have attracted a lot of researches during recent years. However, nowadays applications require higher dynamic range. This study introduces the RNS four-moduli set {2^{2n}, 2^n-1, 2^n+1, 2^{n+1}-1} which is obtained by enhancing the moduli set {2^n-1, 2^n, 2^n+1, 2^{n+1}-1}. This enhancement didn't increase the total speed of RNS arithmetic unit since the critical modulo in both of the moduli sets {2^n-1, 2^n, 2^n+1, 2^{n+1}-1} and {2^{2n}, 2^n-1, 2^n+1, 2^{n+1}-1} is 2^n+1. Besides, an efficient RNS-to-binary converter for the proposed moduli set is designed using a two-level architecture where a previous converter design for subset {2^{2n}, 2^n-1, 2^n+1} is used in the first level and then a two-channel Mixed-Radix Conversion (MRC) algorithm is considered to achieve the final result. Comparison with a recently introduced RNS-to-binary converter for a four-moduli set with the same dynamic range show that the proposed design results in higher speed.

Keywords: Mixed-Radix Conversion (MRC), Residue Number System (RNS), reverse converter

INTRODUCTION

Nowadays, the Residue Number System (RNS) have found considerable applications such as Digital Signal Processing (DSP) and cryptography systems where we need high-speed and low-power hardware implementations of addition, subtraction and multiplications (Soderstrand *et al*., 1986; Cardarilli *et al*., 2007; Conway and Nelson, 2004). The RNS have capability to perform addition, subtraction and multiplication in a fully parallel manner since there is not any carry-propagation between residue digits in RNS. However, other operations such as division, magnitude comparison and sign detection are non-modular processes that are considered as hard RNS operations (Mohan, 2002).

In order to construct an RNS, first some pair-wise relatively prime numbers should be selected to forms the moduli set of the system. The product of these selected numbers determines the range of numbers which can be represented that is called as Dynamic Range (DR). To adapt RNS with other digital systems two converters are needed. First, binary-to-RNS converter transforms the weighted binary number to RNS representation by computing the residue of division of that number to the each modulo of moduli set. Also, RNS-to-binary converter decodes the RNS represented number to its equivalent weighted representation. In contrast to binary-to-RNS converter, the RNS-to-binary converter is so complex. Two famous algorithms to perform RNS-to-binary conversion are Chinese Remainder Theorem (CRT) and Mixed-Radix Conversion (MRC) (Omondi and Premkumar, 2007).

The kind of the moduli set has an important impact on other parts of the RNS system. Due to this the problem of selection of appropriate RNS moduli set attracts many researches during the previous years and as a result some moduli sets have been proposed for different RNS applications. First, the three-moduli sets such as {2^n-1, 2^n, 2^n+1} (Wang *et al*., 2002), {2^n-1, 2^n, 2^{n+1}-1} (Mohan, 2002), {2^n-1, 2^n, 2^{n-1}-1} (Wang *et al*., 2000) and {2^{2n}, 2^n-1, 2^n+1} (Hiasat and Sweidan, 2004) have been considered since the less number of moduli results in low-complexity converters. However, the need for more parallelism lead to introducing four-moduli sets such as {2^n-1, 2^n, 2^n+1, 2^{n+1}-1}, {2^n-1, 2^n, 2^n+1, 2^{n-1}-1} (Cao *et al*., 2005) and {2^n-1, 2^n, 2^n+1, 2^{n+1}+1} (Mohan and Premkumar, 2007). Nowadays, high-performance applications require larger dynamic range. Hence, high DR moduli sets such as {2^n-1, 2^n, 2^n+1, 2^{2n+1}-1}, {2^n-1, 2^n+1, 2^{2n}, 2^{2n}+1} (Molahosseini *et al*., 2010), {2^n-1, 2^n , 2^n+1, 2^{2n}+1} (Cao *et al*., 2003) and {2^n-1, 2^n, 2^n+1, 2^{n+1}-1, 2^{n-1}-1} (Cao *et al*., 2007) were proposed.

In this study, we are going to design efficient reverse converter for the large DR four-moduli set {2^{2n}, 2^n-1, 2^n+1, 2^{n+1}-1} to achieve fast RNS system since this moduli set can provide high dynamic range with relatively the same total RNS arithmetic unit speed than

Corresponding Author: Somayyeh Jafarali Jassbi, Faculty Member of Computer Engineering Department, Science and Research Branch, Islamic Azad University, Tehran, Iran

the moduli set {2^n-1, 2^n, 2^n+1, 2^{n+1}-1}. Because the critical modulo in the proposed set is 2^n+1 and so the enhanced modulo 2^{2n} is not result in increasing the total delay of RNS arithmetic unit. Besides, a two-level RNS-to-binary converter for this moduli set is proposed. The presented converter uses an existing converter for the subset {2^{2n}, 2^n-1, 2^n+1} followed by a two-channel MRC conversion circuit to compute the final weighted number based on the composite set {2^{2n}(2^n-1)(2^n+1), 2^{n+1}-1}.

LITERATURE REVIEW

The residue number system is constructed based on the moduli set {P_1, P_2, …, P_n} where each two moduli P_i and P_j are pair-wise relatively prime (Mohan, 2002; Omondi and Premkumar, 2007). Now, a weighted number X is showed as (x_1, x_2, … , x_n) where:

$$x_i = X \bmod P_i = \left|X\right|_{P_i} \tag{1}$$

In order to convert the RNS number (x_1, x_2, … , x_n) to its equivalent weighted binary number X, the CRT has been used usually. On the other hand, the MRC algorithm can also do the conversion.

Chinese Remainder Theorem (Omondi and Premkumar, 2007): This algorithm describes a method to achieve X by some modulo multiplication and summation as follows:

$$X = \left|\sum_{i=1}^{n} x_i N_i M_i\right|_M \tag{2}$$

Note that M is DR, N^i = $|M^{-1}{}_i|$ P_i is the multiplicative inverse of M_i in modulo P_i and M_i = M/P_i..

Mixed-Radix Conversion (Cao *et al.*, 2005): In contrast to CRT which is a parallel algorithm, the MRC is a sequential algorithm based on the following formulas:

$$X = a_n \prod_{i=1}^{n} P_i + \ldots + a_3 P_2 P_1 + a_2 P_1 + a_1 \tag{3}$$

The coefficients a_is should be calculated using these equations:

$$a_1 = x_1 \tag{4}$$

$$a_2 = \left|(x_2 - a_1)\left|P_1^{-1}\right|_{P_2}\right|_{P_2} \tag{5}$$

…

$$a_n = \left|(((x_n - a_1)\left|P_1^{-1}\right|_{P_n} - a_2)\left|P_2^{-1}\right|_{P_n} - \cdots - a_{n-1})\left|P_{n-1}^{-1}\right|_{P_n}\right|_{P_n} \tag{6}$$

In this study, MRC is used only for two moduli conversion. Hence, the Eq. (6) can be simplified to:

$$X = a_1 + a_2 P_1 = x_1 + P_1 \cdot \left|(x_2 - x_1)\left|P_1^{-1}\right|_{P_2}\right|_{P_2} \tag{7}$$

THE RNS-TO-BINARY CONVERSION ALGORITHM

The numbers of the moduli set {2^{2n}, 2^n-1, 2^n+1, 2^{n+1}-1} are pair-wise relatively prime only for even values of n. Hence, the RNS-to-binary converter for this moduli set works only for even values of n. Besides, Due to the properties of this moduli set a two-level conversion process is selected to achieve an efficient converter. Figure 1 shows the general block diagram of the converter.

The first level relies on an existing RNS-to-binary converter for the subset {2^{2n}, 2^n-1, 2^n+1} (Hiasat and Sweidan, 2004). As described in Hiasat and Sweidan (2004), the formulas for conversion of residue-represented number (x_1, x_2, x_3) to the weighted number Z are as below:

$$Z = x_1 + 2^{2n} Y \tag{8}$$

where,

$$Y = \left|\ell_1 + \ell_2 + \ell'_3 + k\right|_{2^{2n}-1} \tag{9}$$

$$\ell_1 = \underbrace{\bar{x}_{1,2n-1} \cdots \bar{x}_{1,1}\bar{x}_{1,0}}_{2n\ bits} \tag{10}$$

$$\ell_2 = \underbrace{x_{2,0}x_{2,n-1} \cdots x_{2,1}}_{n\ bits}\underbrace{x_{2,0}x_{2,n-1} \cdots x_{2,1}}_{n\ bits} \tag{11}$$

$$\ell'_3 = x_{3,0}\underbrace{\bar{x}_{3,n-1} \cdots \bar{x}_{3,1}\bar{x}_{3,0}}_{n\ bits}\underbrace{x_{3,n-1} \cdots x_{3,2}x_{3,1}}_{n-1\ bits} \tag{12}$$

$$k = \bar{x}_{3,n}\underbrace{0 \cdots 00}_{n-1\ bits}x_{3,n}\underbrace{1 \cdots 11}_{n-1\ bits} \tag{13}$$

This three-modulo converter consists of two $2n$-bit carry-save adders (CSAs) with end-around carries (EACs) followed by a modulo (2^{2n}-1) adder which can be implemented by a $2n$-bit carry-propagate adder (CPA) with EAC (Piestrak, 1994).

The second level uses a two-channel MRC algorithm to combine the result of first level with the fourth residue. First, the required multiplicative inverse is computed using the following lemmas.

Lemma 1: the multiplicative inverse of 2^{2n}(2^n-1)(2^n+1) modulo 2^{n+1}-1 is as follows:

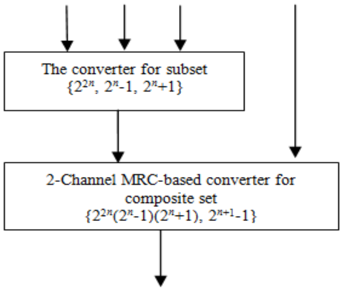

The converter for subset
{2^{2n}, 2^n-1, 2^n+1}
2-Channel MRC-based converter for composite set
{2^{2n}(2^n-1)(2^n+1), 2^{n+1}-1}

Fig. 1: The block diagram of the converter

$$\left|k\cdot 2^{2n}(2^n-1)(2^n+1)\right|_{2^{n+1}-1}=1 \Rightarrow k=\frac{-2^4}{3} \tag{14}$$

Proof: It is clear that:

$$\left|k\cdot 2^{2n}(2^n-1)(2^n+1)\right|_{2^{n+1}-1}=\left|\frac{-2^4}{3}\cdot 2^{2n}(2^{2n}-1)\right|_{2^{n+1}-1}$$
$$=\left|\frac{-1}{3}\cdot 2^{2n+2}(2^{2n+2}.-4)\right|_{2^{n+1}-1}=\left|\frac{-1}{3}\cdot 1.(-3)\right|_{2^{n+1}-1}=1$$

Lemma 2: The value of |1/32|$^{\text{n+1}}$ $_{\text{-1}}$ can be calculated using the following formula where n is even (Cao *et al*., 2005):

$$\left|\frac{1}{3}\right|_{2^{n+1}-1}=\frac{2^{n+2}-1}{3}=\sum_{i=0}^{n/2}2^{2i} \tag{15}$$

Proof: The proof described in Cao *et al*. (2005). Now, consider the moduli set $\{2^{2n}(2^n\text{-}1)(2^n\text{+}1), 2^{n+1}\text{-}1\}$ and X=(Z, x_4). Using the MRC technique (7), X can be computed by this equation:

$$X=Z+2^{2n}(2^{2n}-1)\left|k(x_4-Z)\right|_{2^{n+1}-1} \tag{16}$$

Considering (8), the above equation can be rewritten as:

$$\begin{aligned}X&=x_1+2^{2n}Y+2^{2n}(2^{2n}-1)T\\&=x_1+2^{2n}(Y+2^{2n}T-T)\end{aligned} \tag{17}$$

where,

$$T=\left|k(x_4-Z)\right|_{2^{n+1}-1} \tag{18}$$

Substituting the value of k and Z from (14) and (8), respectively in (18) results in:

$$T=\left|\frac{-2^4}{3}(x_4-x_1+2^{2n}Y)\right|_{2^{n+1}-1}=\left|\frac{2^4}{3}(x_1-x_4-2^{2n}Y)\right|_{2^{n+1}-1} \tag{19}$$

Since, x_1 and Y are 2n-bit numbers, they have to be divided into (n+1)-bit parts. Hence (19) can be written as:

$$T=\left|\frac{2^4}{3}.P\right|_{2^{n+1}-1} \tag{20}$$

where,

$$P=\left|x_1-x_4-2^{2n}Y\right|_{2^{n+1}-1}=\left|P_1+P_2+P_3+P_4+P_5\right|_{2^{n+1}-1} \tag{21}$$

$$P_1=\underbrace{x_{1,n}\ldots x_{1,1}x_{1,0}}_{n+1\,bits} \tag{22}$$

$$P_2=\underbrace{00x_{1,2n-1}\ldots x_{1,n+2}x_{1,n+1}}_{n+1\,bits} \tag{23}$$

$$P_3=\left|-x_4\right|_{2^{n+1}-1}=\underbrace{\bar{x}_{4,n}\ldots\bar{x}_{4,1}\bar{x}_{4,0}}_{n+1\,bits} \tag{24}$$

$$P_4=\left|-2^{2n}(\underbrace{Y_n\ldots Y_1Y_0}_{n+1\,bits})\right|_{2^{n+1}-1}=\left|2^{n-1}(\underbrace{\bar{Y}_n\ldots\bar{Y}_1\bar{Y}_0}_{n+1\,bits})\right|_{2^{n+1}-1} \tag{25}$$
$$=\underbrace{\bar{Y}_1\bar{Y}_0\bar{Y}_n\ldots\bar{Y}_3\bar{Y}_2}_{n+1\,bits}$$

$$P_4=\left|-2^{2n}(\underbrace{00Y_{2n-1}\ldots Y_{n+2}Y_{n+1}}_{n+1\,bits})\right|_{2^{n+1}-1}=\left|2^{n-1}(\underbrace{11\bar{Y}_{2n-1}\ldots\bar{Y}_{n+2}\bar{Y}_{n+1}}_{n+1\,bits})\right|_{2^{n+1}-1} \tag{26}$$
$$=\underbrace{\bar{Y}_{n+2}\bar{Y}_{n+1}11\ldots\bar{Y}_{n+3}\bar{Y}_{n+2}}_{n+1\,bits}$$

Note that two well-known modulo 2^k-1 arithmetic properties which described in Cao *et al*. (2005) are used in deriving (22)-(26). Next, by substituting (15) in (20) we have:

$$T=\left|2^4(2^0+2^2+\ldots+2^n).P\right|_{2^{n+1}-1}=\left|(2^4+2^6+\ldots+2^{n+4}).P\right|_{2^{n+1}-1} \tag{27}$$

This equation can be written as follows:

$$T=\left|P^{(4)}+P^{(6)}+\ldots+P^{(n+4)}\right|_{2^{n+1}-1} \tag{28}$$

where $P^{(i)}$ denotes i-bit circular left shifting of P (Cao *et al*., 2005).

THE HARDWARE ARCHITECTURE OF CONVERTER

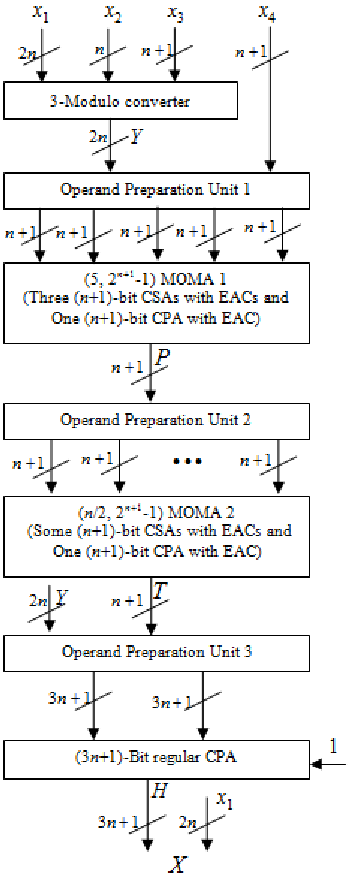
Hardware realization of the proposed RNS-to-binary converter for the moduli set $\{2^{2n}, 2^n\text{-}1, 2^n\text{+}1, 2^{n+1}\text{-}1\}$ consists of two parts. The first part is the circuits for combining the residues x_1, x_2 and x_3 based on the moduli set $\{2^{2n}, 2^n\text{-}1, 2^n\text{+}1\}$. Here, the method of Hiasat and Sweidan (2004) is used to implement this part. Next, (17) should be implemented to achieve the final result. But, before it, the formulas (21) and (28) have to realize. The Eq.(21) requires a (7, 2^{n+1}-1)

Table 1: Details of converter hardware components

Parts	FA	NOT	XOR/AND	XNOR/OR	Delay
Three Moduli Conv.	4n+2	3n+1	n-1	n-1	$(4n+2)t_{FA}$
OPU1	3n+1	–	–	–	tNOT
MOMA1	8n	–	–	–	$(2n+5)t_{FA}$
OPU2	–	–	–	–	–
MOMA2	((n-1)(n+1) /2)	–	–	–	$(2n+l)t_{FA}$
OPU3	–	n+1	–	–	tNOT
CPA	n+1	–	–	2n	$(3n+1)t_{FA}$
Total	((n2-1)/2)+ (16+4)n	4n+2	n-1	3n–1	$(11n+8+l)t_{FA}$ $+2t_{NOT}$

Table 2: Performance comparison

Parts	Hardware requirements	Delay
(Mohan and Premkumar, 2007)	$(9n+5+k)A_{FA}+(6n+1)A_{NOT}+(2n)A_{XNOR/OR}$	$(11.5n+6)t_{FA}$
(Cao *et al*., 2005)	$(n2/2+11n+4)A_{FA}+(3n+2)A_{NOT}$	$(11n+8+l)t_{FA}$
(Molahosseini *et al*., 2010)	$(8n+2)A_{FA}+(n-1)A_{XOR/AND}+(4n+1)A_{XNOR/OR}+(7n+1)A_{NOT}+(n)A_{MUX2\times1}$	$(12n+5)t_{FA}$
Proposed	$((n^2-1)/2)+(16+4)n)A_{FA}+(4n+2)A_{NOT}+(n-1)A_{XOR/AND}+(3n-1)A_{XNOR/OR}$	$(11n+8+l)t_{FA}$



x_1
x_2
x_3
x_4
$2n$
n
$n+1$
$n+1$
3-Modulo converter
$2n$
Y
Operand Preparation Unit 1
$n+1$
$n+1$
$n+1$
$n+1$
$n+1$
$(5, 2^{n+1}-1)$ MOMA 1
(Three $(n+1)$-bit CSAs with EACs and
One $(n+1)$-bit CPA with EAC)
$n+1$
P
Operand Preparation Unit 2
$n+1$
$n+1$
•••
$n+1$
$(n/2, 2^{n+1}-1)$ MOMA 2
(Some $(n+1)$-bit CSAs with EACs and
One $(n+1)$-bit CPA with EAC)
$2n$
Y
$n+1$
T
Operand Preparation Unit 3
$3n+1$
$3n+1$
$(3n+1)$-Bit regular CPA
1
H
$3n+1$
$2n$
x_1
X

Fig. 2: Hardware details of the proposed converter

multi-operand modular adder (MOMA) (Piestrak, 1995) which can be realized by three $(n+1)$-bit CSAs with EACs followed by an $(n+1)$-bit CPA with EAC. Besides, (28) also needs a $(n/2, 2^{n+1}-1)$ MOMA that consists of $(n+1)$-bit CASs with EACs.

Finally, implementation of (17) can be easily done by considering two concatenations. First, since Y is a $2n$-bit number, computation of $Y+2^{2n}T$ needs no hardware. Then the results of this concatenation should be added with two's complement of T plus one to perform the needed subtraction. Therefore, the result of this addition should be concatenated with x_1 to form the weighted number X. Figure 2 shows the details of the proposed converter. Besides, Table 1 describes details of each component. Note that $2n$ full adders (FAs) of the last CPA can be replaced by $2n$ XNOR/OR gates due to the constant bits of one of the operands. Furthermore, t_{FA} is the delay of one FA and l is the number of levels of a CSA tree with $n/2$ inputs.

COMPLEXITY COMPUTATION

Three related studies are selected for comparison. First, two RNS-to-binary converters for the $(4n+1)$-bit DR four-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ (Cao *et al*., 2005; Mohan and Premkumar, 2007) are considered. Second, the recent RNS-to-binary for the $(5n+1)$-bit DR moduli set (Molahosseini *et al*., 2010) is also investigated. Table 2 compares the hardware requirements and conversion delays of these converters. Note that the delay of NOT gates ignored in this Table. It can be seen from this Table that the proposed converter is faster than Molahosseini *et al*. (2010) and also it has relatively the same delay than other two converters. However, the converter design of Molahosseini *et al*. (2010) needs less hardware. It should be noted that the proposed moduli set has $(5n+1)$-bit DR while the moduli set $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ has $(4n+1)$-bit DR. So, the proposed converter can provide higher DR with relatively the same delay than Cao *et al*. (2005) and Mohan and Premkumar (2007).

CONCLUSION

This study presents an efficient RNS-to-binary converter for the new RNS moduli set $\{2^{2n}, 2^n-1, 2^n+1, 2^{n+1}-1\}$. The converter for this set is achieved by two-level converter architecture with better delay compared to a recently introduced RNS-to-binary converter for a $5n$-bit DR moduli set. Also, with the new proposed moduli set the internal RNS arithmetic circuits can be implemented efficiently that can lead to efficient RNS system.

REFERENCES

Cao, B., C.H. Chang and T. Srikanthan, 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 T. Circuits-I, 50(10): 1296-1303.

Cao, B., C.H. Chang and T. Srikanthan, 2007. A residue-to-binary converter for a new five-moduli set. IEEE T. Circuits-I, 54(5): 1041-1049.

Cao, B., T. Srikanthan and C.H. Chang, 2005. 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. Comp. Digit Tech., 152(5): 687-696.

Cardarilli, G.C., A. Nannarelli and M. Re, 2007. Residue number system for low-power DSP applications. Proceeding of 41th Asilomar Conference on Signals, Systems and Computers.

Conway, R. and J. Nelson, 2004. Improved RNS fir filter architectures. IEEE T. Circuits-II, 51(1): 26-28.

Hiasat, A. and A. Sweidan, 2004. Residue-to-binary decoder for an enhanced moduli set. IEE Proc. Comp. Digit. Tech., 151(2): 127-130.

Mohan, P.V.A., 2002. Residue Number Systems: Algorithms and Architectures. Kluwer Academic Publishers, Boston.

Mohan, P.V.A. and A.B. Premkumar, 2007. RNS-to-binary converters for two four-moduli set {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 T. Circuits-I, 54(6): 1245-1254.

Molahosseini, A.S., K. Navi, C. Dadkhah, O. Kavehei and S. Timarchi, 2010. Efficient reverse converter designs for the new 4-moduli sets {2^n–1, 2^n, 2^n+1, 2^{2n+1}–1} and {2^n–1, 2^n+1, 2^{2n}, 2^{2n}+1} based on new CRTs. IEEE T. Circuits-I, 57(4): 823-835.

Omondi, A. and B. Premkumar, 2007. Residue Number Systems: Theory and Implementations. Imperial College Press, London, pp: 296.

Piestrak, S.J., 1994. Design of residue generators and multioperand modular adders using carry-save adders. IEEE T. Comput., 423(1): 68-77.

Piestrak, S.J., 1995. A high speed realization of a residue to binary converter. IEEE T. Circuits-II, 42(10): 661-663.

Soderstrand, M.A., W.K. Jenkins, G.A. Jullien and F.J. Taylor, 1986. Residue Number System Arithmetic: Modern Applications in Digital Signal Processing. IEEE Press Piscataway, NJ, USA.

Wang, W., M.N.S. Swamy, M.O. Ahmad and Y. Wang, 2000. A high-speed residue-to-binary converter and a scheme of its VLSI implementation. IEEE T. Circuits-II, 47(12): 1576-1581.

Wang, Y., X. Song, M. Aboulhamid and H. Shen, 2002. Adder based residue to binary numbers converters for (2^n-1, 2^n, 2^n+1). IEEE T. Signal Proces., 50(7): 1772-1779.