Halmstad University Post-Print

Reverse conversion architectures for signed-digit residue number systems

Andreas Persson and Lars Bengtsson

N.B.: When citing this work, cite the original article.

©2006 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.

Persson A, Bengtsson L. Reverse conversion architectures for signed-digit residue number systems. In: 2006 IEEE International Symposium on Circuits and Systems. IEEE; 2006. p. 4-.

DOI: <http://dx.doi.org/10.1109/ISCAS.2006.1693181>

Copyright: IEEE

Post-Print available at: Halmstad University DiVA
<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-2105>

Reverse Conversion Architectures for Signed-Digit Residue Number Systems

Andreas Persson and Lars Bengtsson
Department of Computer Science and Engineering
Computer Engineering Division
Chalmers University of Technology, Sweden
d00adde@dtek.chalmers.se, labe@ce.chalmers.se

Abstract - This paper presents circuits for conversion from radix-2 Signed-Digit Residue Numbers to binary form. Four reverse converters for combined RNS/SD number systems based on different moduli sets are presented. Implementations are compared with respect to timing, area and Area-Delay products. Finite Impulse Response (FIR) filters are used as reference designs in order to evaluate the performance of RNS/SD processing in a typical DSP block using the suggested moduli sets

1 Introduction

The Residue Number System (RNS) is an integer system capable of supporting high speed concurrent arithmetic. In RNS, an integer is decomposed into a set of residues with shorter binary representations, which can be processed independently and in parallel. The use of the Signed-Digit (SD) number system to represent the residues has been suggested as a way to eliminate carry propagation within RNS arithmetic circuits [1], [2]. The SD number system provides a redundant number representation that facilitates carry-free addition. The use of Signed-Digit representation also implies efficient modulo arithmetic, which helps to simplify crucial RNS operations.

The performance of residue number systems depends heavily on the choice of the moduli set and on the conversion from RNS to binary representation. Many moduli sets and reverse converters have been proposed for RNS systems with residues on 2's complement form [3], [4], [5], [6]. In this paper, we investigate moduli sets and converters for use with Signed-Digit Residue Number Systems (RNS/SD).

2 Background

2.1 The Residue Number System

The basis of an RNS is a set of pairwise prime integers $S = \{m_1, m_2, \dots, m_L\}$, where $\gcd(m_i, m_j) = 1$ for $i \neq j$. The set S is the moduli set and the dynamic range of the system is $[0, M)$, where M is the product of all moduli m_i in S . Any integer X within the dynamic range has a unique RNS representation given by an ordered set of residues

$$X \rightarrow \{x_1, x_2, \dots, x_L\}, \quad x_i = |X|_{m_i}, \quad i = 1, 2, \dots, L,$$

where $|X|_{m_i}$ denotes $X \bmod m_i$.

RNS is a non-weighted number system. If integers A and B have RNS representations $\{a_1, a_2, \dots, a_L\}$ and $\{b_1, b_2, \dots, b_L\}$ respectively, then the RNS representation of $C = A \circ B$ is $\{c_1, c_2, \dots, c_L\}$, where $c_i = |a_i \circ b_i|_{m_i}$ and $\circ$ denotes addition, subtraction or multiplication. Since c_i only depends upon a_i , b_i and m_i , each c_i is computed using its own arithmetic unit, often called a channel.

The reconstruction of X from $\{x_1, x_2, \dots, x_L\}$ is based on the Chinese Remainder Theorem (CRT)

$$X = \left| \sum_{i=1}^L \hat{M}_i \hat{M}_i^{-1} |m_i x_i|_M \right|_M, \quad (1)$$

where $M = \prod_{i=1}^L m_i$, $\hat{M}_i = \frac{M}{m_i}$ and $|\hat{M}_i^{-1}|_{m_i}$ is the multiplicative inverse of $\hat{M}_i$ modulo m_i , such that $|\hat{M}_i \hat{M}_i^{-1}|_{m_i} \equiv 1$.

2.2 The Signed-Digit Number System

The radix-2 Signed-Digit number system has the digit-set $\{\bar{1}, 0, 1\}$, where $\bar{1}$ denotes -1. This yields a redundant number representation. For example, "6" can be represented as $[0110]_{SD}$, $[1\bar{1}10]_{SD}$, or $[10\bar{1}0]_{SD}$. Zero is, however, uniquely represented.

To represent an SD digit, y , two binary bits, y^- and y^+ are required. That is, $y = [y^- y^+]$. The value of an n -digit SD integer $Y = [y_{n-1} \dots y_0]_{SD}$ is

$$\sum_{i=0}^{n-1} y_i^+ \times 2^i - \sum_{i=0}^{n-1} y_i^- \times 2^i. \quad (2)$$

Note that it is possible to represent any integer and its negation with an equal number of digits (unlike the 2's complement number system). The negation of an integer is a very simple operation in the SD number system. The negation of $y = [y^- y^+]$ is $(-y) = [y^+ y^-]$, as can be seen by negating Eq. (2). No logic gates are required for this operation.

By exploiting the redundancy of the Signed Digit number representation, carry propagation is limited to one bit position when adding SD numbers [1], [2]. Consequently, addition is performed in constant time, regardless of operand widths.

3 Moduli Selection in RNS/SD

An important consideration when designing residue number systems is the choice of the moduli set $S = \{m_1, m_2, \dots, m_L\}$. Sets with moduli of the forms 2^n , $2^n - 1$ and $2^n + 1$, known as *low-cost moduli*, are of special interest. Such sets facilitate the use of simplified arithmetic units.

The properties of the SD number system, presented in Section 2.2, helps to further simplify modulo arithmetic for the low-cost moduli. Addition modulo $2^n \pm 1$ is performed using SD adders with end-around-carry logic [1]. Because of the limited carry propagation in SD adders, there is no delay penalty for the end-around-carry operation. Unlike the 2's complement number system, the result of modulo $2^n + 1$ addition is represented using n SD-digits, since results for sums greater than $2^n - 1$ is taken from the negative range.

Modulo multiplication by powers of two relies on shift operations, according to the rules in Eq. Set 3, where $x = [x_{n-1} \dots x_0]_{SD}$ is an n -digit SD integer. These operations are accomplished by wiring connections appropriately.

$$\begin{aligned} |2^a x|_{2^n-1} &= [x_{n-a-1} \dots x_0 x_{n-1} \dots x_{n-a}]_{SD} \\ |2^a x|_{2^n} &= [x_{n-a-1} \dots x_0 0_{n-1} \dots 0_{n-a}]_{SD} \\ |2^a x|_{2^n+1} &= [x_{n-a-1} \dots x_0 (-x_{n-1}) \dots (-x_{n-a})]_{SD} \end{aligned}$$

Eq. Set 3 Rules for multiplication by powers of two.

Five moduli sets, all consisting of low cost moduli have been selected for evaluation in this paper.

$$\begin{aligned} S_1 &= \{2^n - 1, 2^n + 1\} \\ S_2 &= \{2^n - 1, 2^n, 2^n + 1\} \\ S_3 &= \{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\} \\ S_4 &= \{2^{n-1}, 2^{n-1} - 1, 2^n - 1, 2^n + 1\}, n \text{ even integer} \\ S_5 &= \{2^n, 2^n - 1, 2^{n-1} - 1, 2^{n-1} + 1\}, n \text{ odd integer} \end{aligned}$$

Moduli-set-specific decoders for Signed-Digit residue numbers based on sets $S_1 \dots S_5$ are presented in Section 4.

4 RNS/SD Decoders

4.1 Decoder for moduli set S_1

The proposed architecture for decoding SD residues with respect to the $S_1 = \{m_1, m_2\} = \{2^n - 1, 2^n + 1\}$ moduli set is based on the CRT procedure presented in Section 2.1. The two multiplicative inverses required for evaluation of the CRT in eq (1) are both powers of two:

$$\begin{aligned} \hat{M}_1 &= 2^n + 1, & |\hat{M}_1^{-1}|_{m_1} &= 2^{n-1}, \\ \hat{M}_2 &= 2^n - 1, & |\hat{M}_2^{-1}|_{m_2} &= 2^{n-1}. \end{aligned}$$

Eq. (1) is reduced to

$$X = |(2^n + 1)2^{n-1}x_1 + (2^n - 1)2^{n-1}x_2|_{2^{2n}-1}. \quad (4)$$

Since the SD representations of x_1 and x_2 both have digit-length n , Eq. (4) can be computed as the sum of two $2n$ -digit SD vectors, formed by concatenation and rotation according to the rules in Eq. Set 3

$$\begin{aligned} X &= |A + B|_{2^{2n}-1} \\ A &= |(2^{2n-1} + 2^{n-1})x_1|_{2^{2n}-1} = [x_{1_0}x_{1_{n-1}} \dots x_{1_0}x_{1_{n-1}} \dots x_{1_1}]_{SD} \\ B &= |(2^{2n-1} - 2^{n-1})x_2|_{2^{2n}-1} = [x_{2_0}\bar{x}_{2_{n-1}} \dots \bar{x}_{2_0}x_{2_{n-1}} \dots x_{2_1}]_{SD} \end{aligned}$$

where $\bar{x}_i$ denotes $(-x_i)$.

X will be in the range $(-M, M)$, due to the fact that SD modulo adders use the negative range as well as the positive. If the output is required to be in the range $[0, M)$, the correct result is obtained by adding $M = [1_{2n-1} \dots 1_0]$ to X when X is negative. Adding constant ones to an SD integer is a simple operation, as shown in [1]. Carry-look-ahead (CLA) adders are used to obtain the binary representation of X according to Eq. (2). Both X and $X + M$ are decoded to binary form, using two CLAs operating in parallel. The correct value is selected by examining the carry-out bit of the adder for X . The proposed decoder is depicted in Figure 1.

4.2 Decoders for moduli set S_2 and S_3

The sets S_2 and S_3 both have moduli products of the form $2^n(2^a - 1)$, where $a = 2n$ for S_2 and $a = 4n$ for S_3 . The decoding of binary-residue number systems based on these sets have previously been studied, for example in [5], [6]. The proposed RNSD/SD decoders for these moduli sets are based on the New CRT-1, detailed in [7]

$$\begin{aligned} X &= x_1 + m_1 X', \\ X' &= \left| k_1(x_2 - x_1) + k_2 m_2(x_3 - x_2) + \dots \right|_{m_2 \dots m_L}, \end{aligned} \quad (5)$$

where

$$\begin{aligned} |k_1 m_1|_{m_2, m_3 \dots m_L} &\equiv 1 \\ |k_2 m_1 m_2|_{m_3 m_4 \dots m_L} &\equiv 1 \\ &\dots \\ |k_{L-1} m_1 m_2 \dots m_{L-1}|_{m_L} &\equiv 1. \end{aligned} \quad (6)$$

For set S_2 , let $m_1 = 2^n$, $m_2 = 2^n + 1$ and $m_3 = 2^n - 1$. The multiplicative inverses k_1 and k_2 from Eq. (6) are

$$k_1 = 2^n, \quad k_2 = 2^{n-1}.$$

The computation of X' in Eq. (5) is accomplished by adding three $2n$ -digit SD vectors formed by concatenation and rotation.

FIGURE 1

RNS/SD decoder for moduli set S_1 .

FIGURE 2

FIGURE 3

RNS/SD decoder for moduli set S_3 .

FIGURE 4

$$X' = |A + B + C|_{2^{2n-1}},$$

where

$$A = \left| -2^n x_1 - x_2 \right|_{2^{2n-1}} = [\tilde{x}_{1n-1} \dots \tilde{x}_{10} \tilde{x}_{2n-1} \dots \tilde{x}_{20}]_{SD}$$

$$B = \left| (2^{2n-1} + 2^{n-1}) x_2 \right|_{2^{2n-1}} = [x_{20} x_{2n-1} \dots x_{20} x_{2n-1} \dots x_{21}]_{SD}$$

$$C = \left| (2^{2n-1} + 2^{n-1}) x_3 \right|_{2^{2n-1}} = [x_{30} x_{3n-1} \dots x_{30} x_{3n-1} \dots x_{31}]_{SD}.$$

Similarly, rearranging the elements of S_3 , such that $m_1 = 2^n$, $m_2 = 2^{2n} + 1$, $m_3 = 2^n + 1$ and $m_4 = 2^n - 1$, yields attractive multiplicative inverses

$$k_1 = 2^{3n}, \quad k_2 = 2^{n-1}, \quad k_3 = 2^{n-2}.$$

For the S_3 moduli set, X' in eq (5) is the result of the addition of four $4n$ -digit SD vectors

$$X' = |A + B + C + D|_{2^{4n-1}},$$

where

$$\begin{aligned} A &= \left| -2^{3n} x_1 \right|_{2^{4n-1}} \\ &= [\tilde{x}_{1n-1} \dots \tilde{x}_{10} 0_{3n-1} \dots 0_0]_{SD} \end{aligned}$$

$$\begin{aligned} B &= \left| (2^{3n-1} - 2^{n-1}) x_2 \right|_{2^{4n-1}} \\ &= [x_{2n} \dots x_{20} \tilde{x}_{2n-1} \dots \tilde{x}_{20} x_{2n-1} \dots x_{2n+1}]_{SD} \end{aligned}$$

$$\begin{aligned} C &= \left| (-2^{4n-2} + 2^{3n-2} - 2^{2n-2} + 2^{n-2}) x_3 \right|_{2^{4n-1}} \\ &= [\tilde{x}_{31} \tilde{x}_{30} x_{3n-1} \dots x_{30} \tilde{x}_{3n-1} \dots \tilde{x}_{30} x_{3n-1} \dots x_{30} \tilde{x}_{3n-1} \dots \tilde{x}_{32}]_{SD} \end{aligned}$$

$$\begin{aligned} D &= \left| (2^{4n-2} + 2^{3n-2} + 2^{2n-2} + 2^{n-2}) x_4 \right|_{2^{4n-1}} \\ &= [x_{41} x_{40} x_{4n-1} \dots x_{40} x_{4n-1} \dots x_{40} x_{4n-1} \dots x_{40} x_{4n-1} \dots x_{42}]_{SD}. \end{aligned}$$

The result, X , is the concatenation of X' and x_1 . To make sure that the result is in the positive range, $M = [1_{a-1} \dots 1_0 0_{n-1} \dots 0_0]$, is added to negative values of X . Note that this has no effect on the lower n bits of X . Figure 2 and Figure 3 shows the hardware architecture of the RNS/SD reverse converters.

4.3 Decoders for moduli sets S_4 and S_5

The two four-moduli sets

$$S_4 = \{2^{n-1}, 2^{n-1} - 1, 2^n - 1, 2^n + 1\}, n \text{ even integer}$$

$$S_5 = \{2^n, 2^n - 1, 2^{n-1} - 1, 2^{n-1} + 1\}, n \text{ odd integer}$$

can be expressed on a common form as

$$\{m_1, m_2, m_3, m_4\} = \{2^a, 2^a - 1, 2^b - 1, 2^b + 1\}$$

where $a = n - 1$, $b = n$ for S_4 and $a = n$, $b = n - 1$ for S_5 .

The proposed decoders for these sets are SD implementations of the two-level approach to RNS decoding, detailed in [8]. One of the first-level decoders combines residues $|x_1|_{m_1}$ and $|x_2|_{m_2}$ to form a binary residue $|X_1|_{m_1 m_2}$ according to Eq. (5) in Section 4.2. The other first-level decoder combines $|x_3|_{m_3}$ and $|x_4|_{m_4}$ to form $|X_2|_{m_3 m_4}$. This is the CRT decoder from Section 4.1. The second level is a converter for the binary-residue number system with moduli set

$$\{m_1 m_2, m_3 m_4\} = \{2^a(2^a - 1), (2^{2b} - 1)\},$$

as described in [8].

In the last conversion step, a modulo $(2^{2b} - 1)$ adder/scaler is required in order to scale X_1 and X_2 with a precalculated value of the multiplicative inverse $|(m_1 m_2)^{-1}|_{m_3 m_4}$. The decoder for moduli set S_4 is depicted in Figure 4.

5 VLSI Implementation results

The presented decoders have been implemented in VHDL and synthesized using Synopsys Design Compiler and a UMC 0.13 μ m standard CMOS cell library with 8 metal layers and a core voltage of 1.2 Volts.

In order to evaluate the performance of RNS processing using the presented moduli sets, RNS/SD Finite Impulse Response (FIR) filters have been implemented. For an RNS with moduli set $S = \{m_1, m_2, \dots, m_L\}$, the corresponding FIR filter is decomposed into L transposed form subfilters working in parallel. The subfilters has eight filter taps, each consisting of a register, an SD modulo multiplier and an SD modulo adder.

When implementing the FIR filters, pipeline stages were added to the RNS/SD converters to maintain a clock cycle that is determined by the critical path of the filter taps. This introduces an additional latency of two clock cycles for the filters with moduli sets $S_1 \dots S_3$ and four clock cycles for filters with moduli sets S_4 and S_5 .

The designs were compiled for typical operating and wire load conditions and synthesised for three different equivalent (binary) wordlengths (16, 32 and 48 bits). For each moduli set, the parameter n was chosen so that the resulting moduli product was as small as possible, but at least equal to the desired dynamic range. The moduli sets used for VLSI implementation are presented in Table I. Table II presents the synthesis results.

As seen in Table II, the decoder for set S_1 has the smallest area and, for the 16-bit dynamic range, also the shortest delay. For larger dynamic ranges, the decoders for S_2 has the shortest delay, as a result of shorter carry chains in the CLA adders. The decoders for S_4/S_5 have significantly longer delay and larger area, due to the constant multiplier in the second stage of the decoder.

When examining the synthesis results for the RNS/SD FIR filters, we see that the filter using modulo set S_1 suffers from the large wordlength of the residues. The unbalanced residue wordlengths in the filter using moduli set S_3 is also causing degraded performance. The moduli sets S_2 , and S_4/S_5 are well balanced and the use of these sets result in shorter delays. The RNS/SD FIR filter with moduli set S_4/S_5 shows the best Area-Delay products.

6 Conclusions

This article presents architectures for Signed-Digit Residue Numbers System decoding. Four decoders for RNS/SD systems with different moduli sets have been implemented and comparisons have been made with respect to area and timing. It is shown that, out of the presented moduli sets, S_1 and S_2 result in reverse converters with the lowest A*D prod-

ucts. However, considering a typical DSP block (FIR filter), the four-moduli sets S_4 and S_5 results in RNS/SD filters with the best A*D product (at the cost of a higher latency penalty).

References

- [1] A. Lindström, M. Nordseth, L. Bengtsson and A. Omondi. "Arithmetic Circuits Combining Residue and Signed Digit Number Representations" in *Lecture Notes in Computer Science (LNCS)*, vol. 2823, pp. 246-257, September 2003.
- [2] S. Wei and K. Shimizu. "A Novel Residue Arithmetic Hardware Algorithm Using a Signed-Digit Number Representation" in *IEICE Transactions on Information and Systems*, vol. E83-D, no. 12, pp. 2056-2064, December 2000.
- [3] W. Wang, M.N.S. Swamy, M.O. Ahmad and Y. Wang, "The Applications of the New Chinese Remainder Theorems for Three Moduli Sets" in *IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 1, pp. 571-576, May 1999.
- [4] B. Vinnakota and V.V. Bapteswara Rao. "Fast Conversion Techniques for Binary-Residue Number Systems" in *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. CAS-41, no. 12, pp. 927-929, December 1994.
- [5] Y. Wang, X. Song, M. Aboulhamid and H. Shen. "Adder based residue to binary number converters for $(2^n-1, 2^n, 2^n+1)$ " in *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1772-1779, July 2002.
- [6] B. Cao, C. Chang and T. Srikanthan. "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" in *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, no. 10, pp. 1296-1303, October 2003.
- [7] Y. Wang, "Residue-to-Binary Converters Based on New Chinese Remainder Theorems", in *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, No.3, pp. 197-205, March 2000.
- [8] A. Skavantzios and T. Saturated. "Grouped-Moduli Residue Number Systems For Fast Signal Processing" in *ISCAS99, Proceedings of the 1999 IEEE International Symposium on Circuits and Systems*, vol. 3, pp. 478-483, May 1999.

TABLE I Moduli sets used for VLSI implementation.

Moduli Set	16 bits		32 bits		48 bits	
	n	Value	n	Value	n	Value
S_1	8	{255, 257}	16	{65535, 65537}	24	$\{2^{24}-1, 2^{24}+1\}$
S_2	6	{63, 64, 65}	11	{2047, 2048, 2049}	16	{65535, 65536, 65537}
S_3	4	{15, 16, 17, 257}	7	{127, 128, 129, 16385}	10	{1023, 1024, 1025, $2^{20}+1$ }
S_4/S_5	5	{32, 31, 15, 17}	9	{512, 511, 255, 257}	13	{8192, 8191, 4095, 4097}

TABLE II VLSI implementation results.

Moduli Set	RNS/SD Decoders									8-tap Transposed Form RNS/SD Fir Filters								
	Delay [ns]			Area [mm ²]			Area x Delay			Delay [ns]			Area [mm ²]			Area x Delay		
	16 bits	32 bits	48 bits	16 bits	32 bits	48 bits	16 bits	32 bits	48 bits	16 bits	32 bits	48 bits	16 bits	32 bits	48 bits	16 bits	32 bits	48 bits
S_1	3.54	5.57	7.56	0.0024	0.0056	0.0075	0.009	0.031	0.057	2.97	3.97	4.92	0.092	0.33	0.73	0.27	1.31	3.59
S_2	3.57	4.98	6.69	0.0031	0.0057	0.0088	0.011	0.028	0.059	2.87	3.67	3.96	0.074	0.25	0.49	0.21	0.91	1.94
S_3	4.44	5.90	7.61	0.0041	0.0073	0.0107	0.018	0.043	0.081	3.26	3.84	4.72	0.085	0.24	0.46	0.28	0.92	2.17
S_4/S_5	7.95	11.62	13.91	0.0057	0.0153	0.0285	0.045	0.178	0.396	2.79	3.77	3.77	0.072	0.21	0.44	0.20	0.79	1.65