

Number Conversions between RNS and Mixed-Radix Number System Based on Modulo $(2^p - 1)$ Signed-Digit Arithmetic

Shugang Wei

Department of Computer Science, Gunma University
1-5-1, Tenjin-cho, Kiryu, Gunma, Japan
wei@ja4.cs.gunma-u.ac.jp

ABSTRACT

In this paper, new hardware algorithms converting the numbers of a residue number system(RNS) into and from the mixed-radix number system(MRNS) using a radix-two signed-digit (SD) arithmetic circuits are presented. In each residue digit of the RNS, integers $m_i = (2^{p_i} - 1)$ are used as the moduli and the modulo m_i addition and multiplication can be performed by an end-around-carry SD adder and a binary modulo m_i SD adder tree, respectively. Therefore, the modulo m_i addition time is independent of the word length of operands, and the modulo m_i multiplication can be performed in a time proportional to $\log_2 p_i$. An efficient method for calculating a multiplicative inverse number of a modulus is also presented by using the SD arithmetic. By the use of the fast SD arithmetic circuits, number converters of RNS-to-MRNS and MRNS-to-RNS can be implemented with shorter delay time than that using a binary number system.

Categories and Subject Descriptors

B.2.4 [High-Speed Arithmetic]: Algorithms; B.7.1 [Types and Design Styles]: Algorithm implementation in Hardware; C.5.4 [VLSI Systems]:

General Terms

Algorithms

1. INTRODUCTION

Residue number system(RNS) is a non-weighted number system in which the arithmetic with carry-free between residue digits can increase the speed of computations[1]. Various methods of applications of RNS in digital signal processing and error detection have been proposed[2, 3, 4].

Integer $(2^p - 1)$ is well used as a modulus for an RNS, where p is an integer, because the additions modulo $(2^p - 1)$

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SBCCI'05, September 4-7, 2005, Florianópolis, Brazil.
Copyright 2005 ACM 1-59593-174-0/05/0009 ...\$5.00.

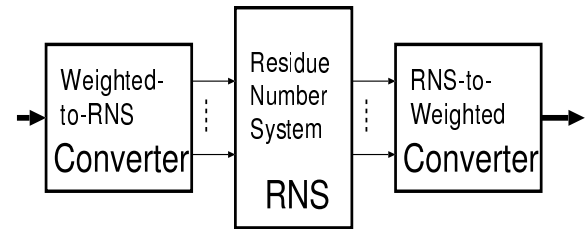


Figure 1: The structure of an RNS with the weighted number converters.

can be implemented by p -bit binary adders[1, 5]. Some modulo $(2^p - 1)$ adders and multipliers have been proposed[5, 6, 7]. However, since these adders and multipliers are constructed based on the ordinary binary arithmetic systems, the carry propagation will arise and the residue addition time is proportional to $\log(p)$ even for the improved adder architectures[8].

It is known that carry propagation is limited to one position during additions of signed-digit (SD) numbers[9]. A number of arithmetic circuits based on SD number systems have been presented[10, 11]. We have introduced a radix-two SD number representation to an RNS for the implementation of high speed residue arithmetic circuits[12, 13]. By using integers $m_i = (2^{p_i} - 1)$ as moduli[12], the modulo m_i addition can be implemented simply using a p_i -digit end-around-carry SD adder, and the modulo m_i multiplier can be implemented with a binary tree structure of the modulo m_i SD adders.

In a computation system, as shown in Fig. 1, the residue numbers should be converted from and into weighted numbers. For some operations, for example, the comparisons can not be performed directly in RNS, and the residue numbers have to be converted into the weighted numbers for a digital-analog conversion.

To convert the RNS to a weighted number system, the Chinese Remainder Theorem is usually applied[1]. In general, a residue arithmetic with a large modulus resulted from the product of the moduli of the RNS is required for the conversion and the conversion circuit may be very complicated. Another method converting the RNS to a weighted number system called mixed-radix number system(MRNS) can be considered, in which the moduli of RNS are associated as the weights[1]. Since only the moduli of RNS are used to perform the residue arithmetic, compact circuit design

and fast number conversion can be expected. Therefore, in this paper, a hardware algorithm converting the RNS to a mixed-radix number system using residue SD arithmetic is presented.

In the following section we mention the basic definitions of the residue number system and the mixed-radix number system. In Sec. 3, a redundant residue number representation is introduced, and the radix-two SD number representation can be used to realize the high speed residue arithmetic. Thus, the residue addition is performed by an SD adder, and the residue multiplication is implemented with a binary SD adder tree. By introducing a Booth recoding method, the residue partial products are reduced to half and the performance of the multiplier can be improved. By using the SD number representation, the multiplicative inversion can be also performed efficiently. In Sec.4, an architecture of the RNS to mixed-radix number converter is proposed. The VLSI implementation using VHDL is also discussed. For the performance evaluation, residue arithmetic circuits and converters using the presented method and a binary number method have been designed. The design and simulation results show that high speed conversion circuits can be achieved by the proposed algorithms by comparing with the binary ones.

2. CONVERSION OF RESIDUE NUMBER SYSTEM TO MIXED NUMBER SYSTEM

2.1 Residue Number System

A residue number system(RNS) has normally a set of relatively prime moduli, $\{m_1, m_2, \dots, m_n\}$, and the residue digit with respect to a modulus m_i is represented by the number set:

$$l_{m_i} = \{0, 1, \dots, m_i - 1\}. \quad (1)$$

An integer A in a value range of $[0, M - 1]$ ($M = \prod_{i=1}^n m_i$) is uniquely represented by the n -tuple (A_1, A_2, \dots, A_n) , where

$$A_i = |A|_{m_i} = A - [A/m_i] \times m_i, \quad (i = 1, 2, \dots, n). \quad (2)$$

In the above equation, each residue digit A_i is defined to be the remainder when A is divided by m_i and $[A/m_i]$ is the integer part of A/m_i .

To simplify residue arithmetic, we use as the moduli a set of positive integers:

$$m_i = 2^{p_i} - 1 = 2^{p_i-1} + 2^{p_i-2} + \dots + 1, \quad (3)$$

where p_i is a positive integer. These numbers can be selected by satisfying the condition that they are relatively prime in pairs. For example, an integer set of $(2^7 - 1, 2^{11} - 1, 2^{13} - 1, 2^{15} - 1, 2^{16} - 1, 2^{17} - 1)$ can be used as the moduli of an RNS. Thus, modulo m_i addition can be simply implemented by one end-around-carry adder. However, when the ordinary binary number system is used for the residue arithmetic, the carry propagation will limit the speed of residue operations.

2.2 Associated mixed-radix number system

Magnitude comparison for the residue number system can be facilitated by converting the given residue representations into a mixed-radix number system associated with the moduli of the RNS. It is a weighted number system, with the representation for a number x given by

$$x = y_n w_n + y_{n-1} w_{n-1} + \dots + y_2 w_2 + y_1 w_1. \quad (4)$$

w_i is the weight of i th digit as follows:

$$w_1 = 1 \quad (5)$$

$$w_i = m_{i-1} m_{i-2} \dots m_1 \quad (6)$$

$$= m_{i-1} w_{i-1}, \quad 2 \leq i \leq n - 1, \quad (7)$$

where m_i is a modulus of the RNS to be converted. Each digit y_i of the mixed-radix number representation is a number in l_{m_i} and satisfying $0 \leq y_i < m_i$. For example, in a mixed-radix number system associated with an RNS having moduli $(m_3, m_2, m_1) = (2^7 - 1, 2^{11} - 1, 2^{13} - 1)$, a number x is represented by (y_3, y_2, y_1) , where

$$x = y_3(2^{11} - 1)(2^{13} - 1) + y_2(2^{13} - 1) + y_1$$

and the digits y_i satisfy

$$0 \leq y_3 < 2^7 - 1,$$

$$0 \leq y_2 < 2^{11} - 1,$$

and

$$0 \leq y_1 < 2^{13} - 1.$$

From (y_3, y_2, y_1) , a residue number modulo m_i associated with the MRNS can be converted by the following relationship.

$$x_i = |x|_{m_i} = |y_i w_i + \dots + y_1 w_1|_{m_i}$$

The conversion from a number represented with (x_3, x_2, x_1) in the RNS to the associated mixed-radix number representation (y_3, y_2, y_1) is calculated by using the following equations[1]:

$$y_1 = |x|_{m_1} = x_1 \quad (8)$$

$$y_2 = |(x_2 - y_1)|_{\frac{1}{m_1}|m_2|_{m_2}} \quad (9)$$

$$y_3 = |((x_3 - y_1)|_{\frac{1}{m_1}|m_3 - y_2})|_{\frac{1}{m_2}|m_3|_{m_3}} \quad (10)$$

where $|\frac{1}{m_1}|_{m_2}$, $|\frac{1}{m_1}|_{m_3}$ and $|\frac{1}{m_2}|_{m_3}$ are the multiplicative inverses of m_1 and m_2 , respectively. From the above equations, the converter can be implemented by a number of modulo m adders and multipliers.

Example 1 : Let a moduli set of an RNS be $(m_3, m_2, m_1) = (2^2 - 1, 2^3 - 1, 2^5 - 1) = (3, 7, 31)$, and $X = 143 = (x_3, x_2, x_1) = (2, 3, 19)$. Then we convert the number representation to the mixed-radix number representation with (y_3, y_2, y_1) as follows:

$$\begin{aligned} y_1 &= 19; \\ y_2 &= |(3 - 19)|_{\frac{1}{31}|7|_7} \\ &= |5 \times 5|_7 \\ &= 4 \\ y_3 &= |((2 - 19)|_{\frac{1}{31}|3 - 4})|_{\frac{1}{7}|3|_3} \\ &= |(1 \times 1 - 4) \times 1|_3 \\ &= 0 \end{aligned}$$

Therefore, $(y_3, y_2, y_1) = (0, 4, 19)$, and $X = 0w_2 + 4w_1 + 19 = 143$, where $w_2 = 7 \times 31 = 217$ and $w_1 = 31$. \square

3. RESIDUE ARITHMETIC BASED ON SIGNED-DIGIT NUMBER REPRESENTATION

3.1 Redundant residue number representation

A residue number x can be represented by a p -digit radix-two SD number representation as follows:

$$x = x_{p-1}2^{p-1} + x_{p-2}2^{p-2} + \cdots + x_0, \\ x_i \in \{-1, 0, 1\} \quad (i = 0, 1, \dots, p-1), \quad (11)$$

which can be denoted as $x = (x_{p-1}, x_{p-2}, \dots, x_0)_{SD}$. In the SD number representation, x has a value in the range of $[-(2^p - 1), 2^p - 1]$. The SD number representation can be considered for implementation of high speed residue arithmetic. However, it is difficult to know if x is in l_m , because of the redundancy of the SD number representation.

To simplify the manipulation of the modular operation in an SD number representation, we give a redundant residue number representation so that each residue digit has the following redundant residue number set:

$$L_m = \{-(2^p - 1), \dots, -1, 0, 1, \dots, 2^p - 1\}, \quad (12)$$

Thus, x must be in L_m when it is expressed in a p -digit SD number representation. (The subscript i is omitted.) Obviously,

$$-x = -(x_{p-1}, x_{p-2}, \dots, x_0)_{SD} \\ = (-x_{p-1}, -x_{p-2}, \dots, -x_0)_{SD}$$

is in L_m .

Definition: Let X be an integer and $m = 2^p - 1$ be a modulus. Then $x = \langle X \rangle_m$ is defined as an integer in L_m . When $|X|_m \neq 0$, x has one of two possible values given by equations

$$x = \langle X \rangle_m = \text{sign}(X)|\text{abs}(X)|_m, \quad (13)$$

where $\text{abs}(X)$ is the absolute value of X , and

$$x = \langle X \rangle_m = \text{sign}(X)|\text{abs}(X)|_m - \text{sign}(X) \times m, \quad (14)$$

respectively; When $|X|_m = 0$, there are three possible values for x , that is, $-m$, 0 and m . \square The

integer set l_m in Eq.(1) is a partial set of L_m . The numbers as the intermediate results calculated in L_m are used for a high speed residue arithmetic system as shown in Fig.1. To convert the residue SD numbers into the weighted numbers, it is necessary to convert the residue numbers from L_m into l_m .

3.2 Residue SD arithmetic circuits

In this paper, the radix-two SD number representation is used for the residue arithmetic. For the same modulo m addition as the above, we have the following algorithm.

[Algorithm 1 (Calculation of $\langle a + b \rangle_m$)] Let a and b be two integers in the p -digit SD number representation, and c_i, z_i and s_i be the intermediate carry, intermediate sum and sum at the i th SD digit ($i = 0, 1, \dots, p-1$), respectively. For each digit, the following two steps are performed.

(ADD1) When $|a_i| = |b_i|$,

$$z_i = 0$$

and

$$c_i = (a_i + b_i)/2;$$

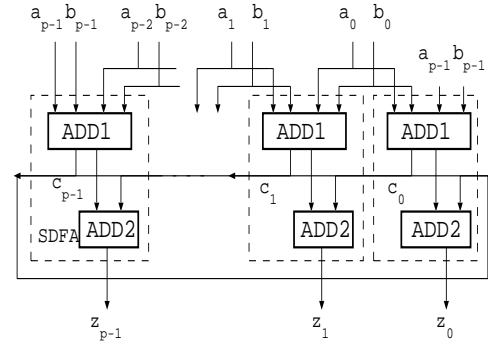


Figure 2: Modulo m SD adder (MSDA).

| | | | | | |
|--------|-------|---|----|----|----|
| | i : 4 | 3 | 2 | 1 | 0 |
| | a : 1 | 0 | -1 | 1 | 1 |
| | b : 1 | 1 | -1 | 0 | 0 |
| (Add1) | | | | | |
| | z : | 0 | 1 | 0 | -1 |
| | c : | 1 | 0 | -1 | 1 |
| | s : | 0 | 0 | 1 | 0 |

Figure 3: Example of modulo m SD addition.

when $|a_i| \neq |b_i|$,

$$z_i = \begin{cases} -(a_i + b_i) & \text{if } a_i + b_i \text{ and } a_{i-1} + b_{i-1} \\ & \text{have the same signs} \\ a_i + b_i & \text{otherwise} \end{cases},$$

and

$$c_i = \begin{cases} a_i + b_i & \text{if } a_i + b_i \text{ and } a_{i-1} + b_{i-1} \\ & \text{have the same signs} \\ 0 & \text{otherwise} \end{cases}.$$

Since $a_i, b_i \in \{-1, 0, 1\}$, $z_i, c_i \in \{-1, 0, 1\}$.

(ADD2) $s_i = z_i + c_{i-1}$.

When $i = 0$, $(i - 1)$ is regarded as $(p - 1)$ in the above equations. Thus,

$$\langle a + b \rangle_m = s = s_{p-1}2^{p-1} + s_{p-2}2^{p-2} + \cdots + s_0. \quad \square$$

In the above algorithm, it is always true that $2c_i + z_i = a_i + b_i$ and z_i and c_{i-1} do not have the same sign so that $s_i = z_i + c_{i-1} \in \{-1, 0, 1\}$. Thus the carry propagations are limited to one digit. A modulo m SD adder can be constructed using p SD full adders (SDFAs) as shown in Fig.2. The blocks ADD1 and ADD2 perform step 1 and step 2 of Algorithm 1, respectively.

Example 2 : Let $p = 5$, $a = (1, 0, -1, 1, 1)_{SD}$ and $b = (1, 1, -1, 0, 0)_{SD}$, so that $m = 31$, $a = 15$ and $b = 20$. Fig.3 illustrates the calculation of $\langle a + b \rangle_{31}$ using the above algorithm. The result is $\langle 15 + 20 \rangle_{31} = 4$. \square

The calculation of $\langle a - b \rangle_m$ can be realized by replacing b with $-b$ in the above algorithm.

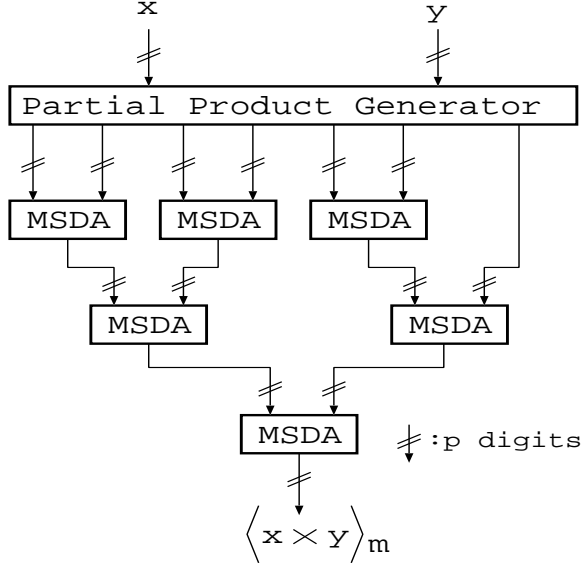


Figure 4: Modulo m SD multiplier(MSDM).

3.3 Residue SD multiplication with a Booth recoding

To calculate $\langle x \times y \rangle_m$, where x and y are integers in the p -digit radix-two SD number representation, we have

$$\begin{aligned} \langle x \times y \rangle_m &= \langle \sum_{i=0}^{p-1} \langle y_i 2^i \times (x_{p-1} 2^{p-1} + x_{p-2} 2^{p-2} + \dots + x_0) \rangle_m \rangle_m \\ &= \langle \sum_{i=0}^{p-1} pp_i \rangle_m, \end{aligned} \quad (15)$$

where

$$pp_i = \langle y_i 2^i \times (x_{p-1} 2^{p-1} + x_{p-2} 2^{p-2} + \dots + x_0) \rangle_m \quad (16)$$

denotes as a partial product. Since $y_i \in \{-1, 0, 1\}$,

$$\begin{aligned} pp_i &= y_i \langle 2^i \times (x_{p-1} 2^{p-1} + x_{p-2} 2^{p-2} + \dots + x_0) \rangle_m \\ &= y_i \times sx_i, \end{aligned} \quad (17)$$

where

$$sx_i = \langle 2^i (x_{p-1} 2^{p-1} + x_{p-2} 2^{p-2} + \dots + x_0) \rangle_m. \quad (18)$$

Since $\langle 2^{p+i} \rangle_m = \langle \langle 2^p \rangle_m \times 2^i \rangle_m = 2^i$, the modular shift operation is performed as follows:

$$\begin{aligned} sx_i &= \langle 2^i \times (x_{p-1} 2^{p-1} + x_{p-2} 2^{p-2} + \dots + x_0) \rangle_m \\ &= x_{p-i-1} 2^{p-1} + x_{p-i-2} 2^{p-2} \\ &\quad + \dots + x_0 2^i + x_{p-1} 2^{i-1} + \dots + x_{p-i} \end{aligned} \quad (19)$$

Therefore, a modulo m multiplication can be implemented by calculating Eqs.(18), (17) and (15) to obtain partial products and the sum of the partial products.

Algorithm 2 ($\langle x \times y \rangle_m$):

Let x and y be two p -digit radix-2 SD numbers.

- (1) Calculate the residue of i -digit shifted numbers in parallel for $i = 0, 1, \dots, p-1$,

$$sx_i = \langle 2^i \times x \rangle_m. \quad (20)$$

- (2) Multiply sx_i by y_i to obtain the partial product, pp_i ($i = 0, 1, \dots, p-1$), shown in Eq.(17).

$$pp_i = y_i \times sx_i$$

- (3) Calculate the modulo m sum of these partial products by performing Algorithm 1 repeatedly.

$$\langle x \times y \rangle_m = \langle pp_0 + pp_1 + \dots + pp_{p-1} \rangle_m$$

□

A binary tree of the modulo m SD adders(MSDAs) can be constructed as shown in Fig.4 and the multiplication time is proportional to $\log_2 p$.

3.4 Multiplicative inverse with SD number representation

As shown in Eqs.(9) and (10), the residue multiplicative inverses are used for the conversion calculations from an RNS to a mixed-radix number system. In this paper, we present an efficient method to calculate the multiplicative inverses by using the radix-two SD number representation. The following properties are useful for the calculations of the inverses.

Property 1: Let $m = 2^p - 1$ and k be a positive integer. Then

$$\langle 2^k \rangle_m = 2^{\langle k \rangle_p}. \quad (21)$$

When $k = n \times p$, specially,

$$\langle 2^{n \times p} \rangle_m = 1. \quad (22)$$

□

Property 2: Let m_i and m_j be two moduli in an RNS. When $|m_i|_{m_j} = m'_i$,

$$\langle \frac{1}{m_i} \rangle_{m_j} = \langle \frac{1}{m'_i} \rangle_{m_j}.$$

□

The above property is based on the following obvious fact:

$$\langle (n \times m_j + m'_i) \langle \frac{1}{m'_i} \rangle_{m_j} \rangle_{m_j} = 1,$$

where n is an integer and $n \geq 0$. When $m_i = 2^{p_i} - 1$ and $m_j = 2^{p_j} - 1$, $m'_i = 2^{\lfloor p_i/p_j \rfloor} - 1$. Using the above properties, we have

$$\begin{aligned} 1 &= \langle 2^{n \times p_j} - (2^{n \times p_j} - 1) \rangle_{m_j} \\ &= \langle -(2^{n \times p_j} - 2) \rangle_{m_j} \\ &= \langle -2(2^{n \times p_j - 1} - 1) \rangle_{m_j}. \end{aligned}$$

Thus, 1 can be expressed as an $(n \times p_j)$ -digit SD number representation, that is, $1 = \langle (-1, -1, \dots, -1, -1, 0)_{SD} \rangle_{m_j}$.

If $(2^{n \times p_j - 1} - 1)/m'_i$ is an integer, then

$$\begin{aligned} \langle \frac{1}{m'_i} \rangle_{m_j} &= \langle \frac{(-1, -1, \dots, -1, -1, 0)_{SD}}{(1, 1, \dots, 1)_{SD}} \rangle_{m_j} \\ &= \langle (-1, 0, \dots, 0, -1, \dots, 0, \dots, 0, -1, 0)_{SD} \rangle_{m_j}. \end{aligned}$$

In the above equation, the numbers with an underline have a $(\lfloor p_i/p_j \rfloor)$ -digit word length. Based on the above discussion and properties, a multiplicative inverse number, $\langle \frac{1}{m_i} \rangle_{m_j} = \langle \frac{1}{m'_i} \rangle_{m_j}$, can be calculated efficiently by using the SD arithmetic.

Multiplicative Inversion : Let $m_i = 2^{p_i} - 1$ and $m_j = 2^{p_j} - 1$ be two moduli in an RNS.

- (1) Calculate $p'_i = |p_i|_{p_j}$. Thus, m'_i is expressed as a (p'_i) -digit SD number representation,

$$m'_i = (1, 1, \dots, 1)_{SD}.$$

- (2) Express 1 as an $(n \times p_j)$ -digit SD number representation,

$$\langle 1 \rangle_{m_j} = (-1, -1, \dots, -1, 0)_{SD},$$

and $n \times p_j - 1 = k \times p'_i$, where k is an integer. Thus, $\frac{1}{m'_i}$ is expressed as an $(n \times p_j)$ -digit SD number with k '1's.

- (3) Divide the SD number into n p_j -digit SD numbers and perform Algorithm 1 repeatedly for the modulo m_j sum of them.

□ **Example 3 :** Let $m_i = 2^{p_i} - 1 = 2^7 - 1 = 127$ and $m_j = 2^{p_j} - 1 = 2^4 - 1 = 15$.

$$\begin{aligned} \langle \frac{1}{m_i} \rangle_{15} &= \langle \frac{(-1, -1, \dots, 1, -1, 0)_{SD}}{(1, 1, 1)_{SD}} \rangle_{15} \\ &= \langle (0, 0, -1, 0, 0, -1, 0, 0, -1, 0, 0, 0, -1, 0, 0, -1, 0, 0, -1, 0)_{SD} \rangle_{15}. \end{aligned}$$

Thus,

$$\begin{aligned} \langle \frac{1}{127} \rangle_{15} &= \langle (0, 0, -1, 0, 0, -1, 0, 0, -1, 0, 0, -1, 0, 0, -1, 0, 0, -1, 0, 0, -1, 0)_{SD} \rangle_{15} \\ &= \langle (0, 0, -1, 0)_{SD} + (0, -1, 0, 0)_{SD} \\ &\quad + (-1, 0, 0, -1)_{SD} + (0, 0, -1, 0)_{SD} \rangle_{15} \\ &= (0, 0, -1, 0)_{SD} \end{aligned}$$

Therefore,

$$\langle \frac{1}{127} \rangle_{15} = -2. \quad \square$$

4. CONVERTER OF RNS TO MIXED-RADIX NUMBER SYSTEM USING MODULO M SD ARITHMETIC

4.1 Structure of the proposed converter

The idea to implement the high speed conversion is to apply the proposed residue SD arithmetic. Since the carry propagation is limited to one position for the modulo m addition and the modulo m multiplier can be constructed as an SD adder tree, the high speed arithmetic both in RNS and converters can be achieved by using the residue SD arithmetic circuits. The conversion with three-moduli from MRNS to RNS can be performed by the the SD arithmetic operations: Let $m_i = 2^{p_i} - 1$ and $i = 1, 2, 3$. Then (x_3, x_2, x_1) in RNS, can be calculated by

$$\begin{aligned} x_1 &= \langle y_3 m_2 m_1 + y_2 m_1 + y_1 \rangle_{m_1} \\ &= y_1 \\ x_2 &= \langle y_2 m_1 + y_1 \rangle_{m_2} \\ &= \langle y_2 2^{p_1} - y_2 + y_1 \rangle_{m_2} \\ x_3 &= \langle y_3 2^{p_1 + p_2} - y_3 2^{p_2} - y_3 2^{p_1} + y_3 + y_2 2^{p_1} - y_2 + y_1 \rangle_{m_3} \end{aligned}$$

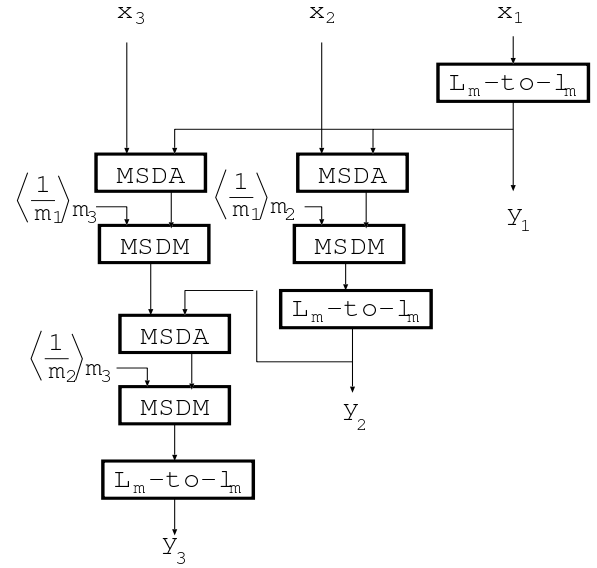


Figure 5: Converter using SD representations with 3 moduli.

The above calculations for the conversion from MRNS to RNS are implemented by shifts and residue SD additions and the residue numbers are represented in the SD number representation. Figure 5 shows a structure of the converter of a three-moduli RNS to a mixed-radix number system to implement the above calculations. The final output y_i is obtained by the circuit of $L_m - to - l_m$ as follows: If all digits of the input data y'_i are 1 ($y'_i = m_i$), then all digits of y'_i are set to 0; Otherwise the output is obtained by performing $\langle y_i = y'_i + m_i \rangle_{m_i}$. For example, when $m_i = 31$ and $y'_i = (-1, 0, 1, 1, 0)_{SD}$, $y_i = \langle (-1, 0, 1, 1, 0)_{SD} + (1, 1, 1, 1, 1)_{SD} \rangle_{31} = (1, 0, 1, 0, 1)$; When $y'_i = (1, 0, -1, -1, 0)_{SD}$, $y_i = \langle (1, 0, -1, -1, 0)_{SD} + (1, 1, 1, 1, 1)_{SD} \rangle_{31} = (0, 1, 0, 1, 0)$. Thus, the functional block $L_m - to - l_m$ also converts an SD number into a binary number.

4.2 On VLSI implementation

In this paper, VHDL is used to describe the residue arithmetic circuits for the implementation of the presented algorithms. We specify a binary representation for a radix-two signed-digit $a_i = [a_i(1)a_i(0)]$, where $a_i(1)$ is the sign and $a_i(0)$ is the absolute value of a_i . Thus, a p -digit radix-two SD number a is represented by a vector with $2p$ -bit length. For example, $(1, 0, 0, -1)_{SD} = [01000011]$. Using the binary representation, modular SD arithmetic operations are described and simulated. From the VHDL description codes, the simulation and the logic circuit synthesis are performed by using VHDL synthesis software tool.

We suppose that the VLSI implementation is based on a gate array IC, because ASIC design on a gate array is a popular VLSI implementation method. In Table 1, the circuit design resulting from modulo m SD arithmetic circuits and that from the binary ones are listed for the performance comparisons. In our experiments, the delay time is obtained by simulation under the condition of $1\mu m$ CMOS technology. The proposed converter using the SD arithmetic has a 3-moduli set and much faster than the binary one.

Table 1: Performance of modulo $2^p - 1$ arithmetic circuits.

| Circuit | Number of Gates | | Delay Time(ns) | |
|------------|-----------------|--------|-----------------|--------|
| | SD | Binary | SD | Binary |
| $p = 11$ | | | | |
| Adder | 363 | 155 | 5.33 | 18.29 |
| Multiplier | 1,815 | 1505 | 16.92 | 27.43 |
| Converter* | 4,356 | 3,612 | 57.61 | 109.08 |

* 3-moduli: $(2^7 - 1, 2^{11} - 1, 2^{13} - 1)$

5. CONCLUSIONS

High speed computations can be performed based on the assumption that input and output data of the residue arithmetic circuits are in the residue SD number form, because some computing system applications, such as digital filtering, require repeated calculations of sums of products before the final results are obtained. For integration with conventional binary systems, efficient conversion circuit has been presented.

6. 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. Sonderstrand, W. K. Jendins, G. A. Junllien, and F. J. Taylor,"Residue Number System Arithmetic: Modern Applications in Digital Signal Processing," IEEE Press, New York, 1986.
- [3] D. Mandelbam : "Error correction in residue arithmetic," IEEE Trans. Comput., Vol.C-21, pp.538-545,June 1972.
- [4] F.Barsi and P.Maestrini: "Error correcting properties of redundant residue number systems,"IEEE Trans. Comput., Vol.C-22, pp.307-315,March 1973.
- [5] D.P. Agrawal and T.R.N.Rao,"Modulo $(2^n + 1)$ arithmetic logic,"IEE J. Electronic Circuits and Systems, Vol.2, pp. 186-188, Nov. 1978.
- [6] F.J.Taylor,"A VLSI residue arithmetic multiplier," IEEE Trans. Comput., Vol.C-31, pp.540-546,June 1982.
- [7] A.Hiasat,"New memoryless, mod $(2^n \pm 1)$ residue multiplier," Electron. Lett., Vol.28, No.3, pp.314-315,Jan. 1992.
- [8] L.Kalampoukas, D.nikolos, C.Efstathiou, H.T.Vegos and J. Kakamatianos, "High-Speed Parallel-Prefix Modulo $2^2 - 1$ Adders," IEEE Trans. Comp., vol.49, no.7, pp.673-680, July 2000.
- [9] A.Avizienis,"Signed-digit number representations for fast parallel arithmetic,"IRE Trans. Elect. Comput., EC-10,pp.389-400, Sept. 1961.
- [10] N.Takagi,H.Yasuura and S.Yajima,"High-Speed VLSI multiplication algorithm with a redundant binary addition tree," IEEE Trans. Comput., Vol. C-34, pp.789-796, Sept. 1985.
- [11] C.N.Lyu and D.W.Matula,"Redundant binary booth recoding," Proc. of IEEE 12th Symp.Comput.Arith.,pp.50-57, 1995.
- [12] S.Weï and K.Shimizu,"Modulo $2^p - 1$ arithmetic hardware algorithm using signed-digit number representation," Trans. IEICE. INF. & SYST. Vol.E79-D, No.3, pp. 242-246, March 1996.
- [13] S.Weï and K.Shimizu,"A Novel Residue Arithmetic Hardware Algorithm Using a Signed-Digit Number Representation", IEICE Trans.INF. & SYST., Vol.E83-D, No.12, pp. 2056-2064, Dec. 2000.
- [14] C.Efstathiou, H.T. Vergos and D.Nikolos, "Modified Booth Modulo $2^n - 1$ Multipliers," *IEEE Trans. on comp.*, vol.53, no.3, pp. 370-374, March 2004.