FACTA UNIVERSITATIS (NIŠ)
SER.: ELEC. ENERG. vol. 24, no. 1, April 2011, 89-103

Reverse Convertor Design for the 4-Moduli Set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ Based on the Mixed-Radix Conversion

Negovan Stamenković and Bojan Jovanović

Abstract: The residue number system (RNS) is an integer system capable of supporting high speed concurrent arithmetic. One of the most important consideration when designing RNS system is reverse conversion. The reverse converter for recently proposed for the four-moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ is based on new Chinese remainder theorems II (New CRT-II) [6]. This paper presents an alternative architecture derived by Mixed-Radix conversion for this four-moduli set. Due to the using simple multiplicative inverses of the proposed moduli set, it can considerably reduce the complexity of the RNS to binary converter based on the Mixed-Radix conversion. The hardware architecture for the proposed converter is based on the adders and subtractors, without the needed ROM or multipliers.

Keywords: Computer arithmetic, residue number system, reverse converter, mixed-radix conversion, four-moduli set.

1 Introduction

It is well known that the Residue Number System (RNS) architectures for the digital signal processing are typically composed of three major parts: a binary to RNS converter for converting the weighted number to residue representations, an arithmetic unit containing modular adder, subtracter and multiplier, and a RNS to binary converter for transforming the residues into its equivalent weighted binary representation [4, 7]. Among these, residue-to-binary converter is the most complex part of any RNS architecture which should be efficiently implemented to prevent

Manuscript received on January 8, 2011.

N. Stamenković is with Faculty of Natural Science, 28220 Kosovska Mitrovica, Lole Ribara 29, Serbia (e-mail: negovanstamenkovic@gmail.com). B. Jovanović is with University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18115 Niš, Serbia (e-mail: bojan.jovanovic@elfak.ni.ac.rs).

the performance degradation of the overall RNS system. The moduli set choice is also an important issue since the complexity and the speed of the resulting conversion structure depends on the chosen moduli set [9]. So, the challenges of the RNS system design lie in the choice of the moduli set and in the residue to binary conversion.

The dynamic range of an RNS system is defined in terms of product of the moduli, and it denotes the interval of integers, which can be uniquely presented in RNS. The larger dynamic range can be realized by using four moduli set [5] or by using larger value for n in three power-of-two moduli set [2]. It should be noted that as the number of moduli in the set increase, the complexity of the RNS will increase. Thus, the RNS systems based on four moduli set are more complex than those based on three-moduli set.

An important concern for reverse converter design is the selection of an appropriate conversion algorithm. The algorithms of reverse conversion are mainly based on the Chinese remainder theorem (CRT), mixed-radix conversion (MRC) and the new Chinese remainder theorems (New CRTs) [10]. Among these, New CRTs has simple computations which can be efficiently realized in hardware. For these cases Molahoessini et al. [6] recently proposed reverse converter based on the new CRT for the four moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$ which has a sufficient dynamic range ($5n$).

In this paper, we propose the residue to binary converter for four moduli set, proposed in [6], based on the mixed-radix conversion. First, we proposed simple values for multiplicative inverse which leads to simpler hardware. Second, we reduced hardware by using borrow-save-subtractor instead of carry-save-adder with end-around-carry, and by using proposed new modular subtractor which avoids double presentation of zero.

The paper is organized as follows: in Section 2, we introduce the necessary background; the proposed improvements are presented in Section 3. Section 4 provides hardware implementation and simulation, and the last section is Conclusion.

2 Background

A residue number system (RNS) is defined in terms of a relatively-prime moduli set $\{m_1, m_2, \ldots, m_k\}$ that is $\gcd(m_i, m_j) = 1$ for $i \neq j$ [1, 8]. The greatest common divisor (gcd) for a pair of numbers (a,b), can be calculated by the well known Euclidian algorithm. A binary number X can be represented in the defined residue number system as a set of n smaller integers $X = (x_1, x_2, \ldots, x_k)$, where

$$x_i = \langle X \rangle_{m_i}, \quad 0 \leq x_i < m_i \tag{1}$$

and $\langle X \rangle_{m_i}$ denotes the residue of X modulo m_i. This representation is unique for any integer X in the range $[0, M-1]$, where $M = m_1 m_2 \cdots m_k$ is the dynamic range of the moduli set $\{m_1, m_2, \ldots, m_k\}$. Modulo (2^n-1) of a negative number is accomplished by subtracting this number from (2^n-1). This is equivalent to taking one's complement of the number.

A large number can thus be represented by several smaller numbers, thereby facilitating big word-length operations to be realized as several small word-length operations. The addition, subtraction, and multiplication operations can thus be performed quite efficiently. The division, sign detection, and magnitude comparison are time consuming in RNSs.

The Chinese Remainder Theorem (CRT) and mixed-radix conversion (MRC) are generally used to perform the residue to binary conversion, that is to convert the residue number $(x_1, x_2, \ldots, x_k)$ into the binary number X,

The binary number X is computed by

$$X = \left\langle \sum_{i=1}^{k} \langle x_i N_i \rangle_{m_i} M_1 \right\rangle_M \tag{2}$$

where $M_i = M/m_i$ and $N_i = \langle M_i^{-1} \rangle_{m_i}$ is the multiplicative inverse of M_i modulo m_i. The main drawback of this approach is that it requires multiplication by the M_i's and modulo M operations (which M is large number).

The number X can be computed by

$$X = a_k \prod_{i=1}^{k-1} M_i + \cdots + a_3 m_1 m_2 + a_2 m_1 + a_1 \tag{3}$$

where a_is are called the mixed-radix digits (MRD) and they can be obtained from the residues by [8]

$$\begin{aligned} a_k = &\big\langle (\cdots ((x_k - a_1)c_{1,k} - a_2)c_{2,k} - \cdots \\ &- a_{k-1})c_{k-1,k} \big\rangle_{m_k} \end{aligned} \tag{4}$$

where $c_{i,j}$ for $1 \le i \le j < 3$ is the multiplicative inverse of m_i modulo m_j, or $\langle c_{ij} \times m_i \rangle_{m_j} = 1$, for $k > 1$ and $a_1 = x_1$. For MRDs a_i, $0 \le a_i < m_i$, any positive number in interval $[0, M-1]$ is uniquely represented. From (3), it is also seen that the digit a_k is the most significant digit. It is shown that the Mixed Radix Conversion is a strictly sequential process. There is no need for the final modulo reduction.

3 MRD to Binary Conversion

Suppose that we have residue number (x_1, x_2, x_3, x_4), $0 \le x_i < m_i$, for the moduli set of length four $\{m_1, m_2, m_3, m_4\}$ and by substituting Eq. (3) we obtain the following

expresion:

$$X = a_1 + a_2 m_1 + a_3 m_1 m_2 + a_4 m_1 m_2 m_3 \tag{5}$$

were $[1, m_1, m_1m_2, m_1m_2m_3]$ is mixed-radix vector of mixed-radix system. In (5), a_1, a_2, a_3 and a_4 are represented as a sequential algorithm

$$\begin{aligned} a_1 &= x_1 \\ a_2 &= \langle (x_2 - a_1)c_{12} \rangle_{m_2} \\ a_3 &= \langle ((x_3 - a_1)c_{13} - a_2)c_{23} \rangle_{m_3} \\ a_4 &= \langle (((x_4 - a_1)c_{14} - a_2)c_{24} - a_3)c_{34} \rangle_{m_4} \end{aligned} \tag{6}$$

If the mixed-radix digits are given, any number in the interval $[0, M-1]$ can be uniquely represented. The well known mixed-radix conversion algorithm for four moduli set, according to H. L. Garner [1], is displayed in Figure 1.

Definition 1 *Digits in the residue number system have no ordering significance. In residue addition, subtraction, and multiplication, any particular digit of the resultant depends solely on the corresponding digits of its operands. However, Residue to Mixed-Radix Conversion depends on the digit ordering as shown in (3). Further, mixed-radix digits ordering depends on the moduli set ordering. Due to this reason we define the form of moduli set: the order of modules in the residue number system. For example, assuming four moduli* $2^n - 1$, 2^n, $2^n + 1$, $2^{2n+1} - 1$ *we define the first form of moduli set in ascending order, second form* $2^n - 1$, 2^n, $2^{2n+1} - 1$, $2^n + 1$, *and so on. A set of four modules has twentyfour forms. Finally, the twentyfourth form is a set of modules in descending order. Thus, the modulo at first position is* m_1, *at second position is* m_2, *at third position is* m_3, *and at fourth position is* m_4. ■

The multiplicative inverse for all twenty-four forms of given moduli set are shown in Table 1.The nineteenth form of given moduli set provides the best solution for c_{ij}:

$$\begin{aligned} c_{12} &= -1, \quad & c_{23} &= -1, \\ c_{13} &= 1, \quad & c_{24} &= 1, \\ c_{14} &= 1, \quad & c_{34} &= 2^{n-1}. \end{aligned} \tag{7}$$

It can be seen that the twenty-first form of moduli set also provides a good solution.

Using the nineteenth form of given moduli set mixed-radix digits can be represented as

$$\begin{aligned} a_1 &= x_1 \\ a_2 &= \langle (a_1 - x_2) \rangle_{2^n} \\ a_3 &= \langle (a_2 - (x_3 - a_1)) \rangle_{2^n+1} \\ a_4 &= \langle (((x_4 - a_1) - a_2) - a_3)2^{n-1} \rangle_{2^n-1} \end{aligned} \tag{8}$$

Table 1. Multiplicative inverse for moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$

Forma	m_1	m_2	m_3	m_4	c_{12}	c_{13}	c_{14}	c_{23}	c_{24}	c_{34}
1	2^n-1	2^n	2^n+1	$2^{2n+1}-1$	-1	2^{n-1}	$-2^{n+1}-2$	-1	2^{n+1}	$-2^{n+1}+2$
2	2^n-1	2^n	$2^{2n+1}-1$	2^n+1	-1	$-2^{n+1}-2$	2^{n-1}	2^{n+1}	-1	1
3	2^n-1	2^n+1	2^n	$2^{2n+1}-1$	2^{n-1}	-1	$-2^{n+1}-2$	1	$-2^{n+1}+2$	2^{n+1}
4	2^n-1	2^n+1	$2^{2n+1}-1$	2^n	2^{n-1}	$-2^{n+1}-2$	-1	$-2^{n+1}+2$	1	-1
5	2^n-1	$2^{2n+1}-1$	2^n+1	2^n	$-2^{n+1}-2$	2^{n-1}	-1	1	-1	1
6	2^n-1	$2^{2n+1}-1$	2^n	2^n+1	$-2^{n+1}-2$	-1	2^{n-1}	-1	1	-1
7	2^n	2^n-1	2^n+1	$2^{2n+1}-1$	1	-1	2^{n+1}	2^{n-1}	$-2^{n+1}-2$	$-2^{n+1}+2$
8	2^n	2^n-1	$2^{2n+1}-1$	2^n+1	1	2^{n+1}	-1	$-2^{n+1}-2$	2^{n-1}	1
9	2^n	2^n+1	2^n-1	$2^{2n+1}-1$	-1	1	2^{n+1}	$-2^{n-1}+1$	$-2^{n+1}+2$	$-2^{n+1}-2$
10	2^n	2^n+1	$2^{2n+1}-1$	2^n-1	-1	2^{n+1}	1	$-2^{n+1}+2$	$-2^{n-1}+1$	1
11	2^n	$2^{2n+1}-1$	2^n+1	2^n-1	2^{n+1}	-1	1	1	1	$-2^{n-1}+1$
12	2^n	$2^{2n+1}-1$	2^n-1	2^n+1	2^{n+1}	1	-1	1	1	2^{n-1}
13	2^n+1	2^n	2^n-1	$2^{2n+1}-1$	1	$-2^{n-1}+1$	$-2^{n+1}+2$	1	2^{n+1}	$-2^{n+1}-2$
14	2^n+1	2^n	$2^{2n+1}-1$	2^n-1	1	$-2^{n+1}+2$	$-2^{n-1}+1$	2^{n+1}	1	1
15	2^n+1	2^n-1	2^n	$2^{2n+1}-1$	$-2^{n-1}+1$	1	$-2^{n+1}+2$	-1	$-2^{n+1}-2$	2^{n+1}
16	2^n+1	2^n-1	$2^{2n+1}-1$	2^n	$-2^{n-1}+1$	$-2^{n+1}+2$	1	$-2^{n+1}-2$	-1	-1
17	2^n+1	$2^{2n+1}-1$	2^n-1	2^n	$-2^{n+1}+2$	$-2^{n-1}+1$	1	1	-1	-1
18	2^n+1	$2^{2n+1}-1$	2^n	2^n-1	$-2^{n+1}+2$	1	$-2^{n-1}+1$	-1	1	1
19	$2^{2n+1}-1$	2^n	2^n+1	2^n-1	-1	1	1	-1	1	$-2^{n-1}+1$
20	$2^{2n+1}-1$	2^n	2^n-1	2^n+1	-1	1	1	1	-1	2^{n-1}
21	$2^{2n+1}-1$	2^n+1	2^n	2^n-1	1	-1	1	1	$-2^{n-1}-1$	1
22	$2^{2n+1}-1$	2^n+1	2^n-1	2^n	1	1	-1	$-2^{n-1}+1$	1	-1
23	$2^{2n+1}-1$	2^n-1	2^n+1	2^n	1	1	-1	2^{n-1}	-1	1
24	$2^{2n+1}-1$	2^n-1	2^n	2^n+1	1	-1	1	-1	2^{n-1}	-1

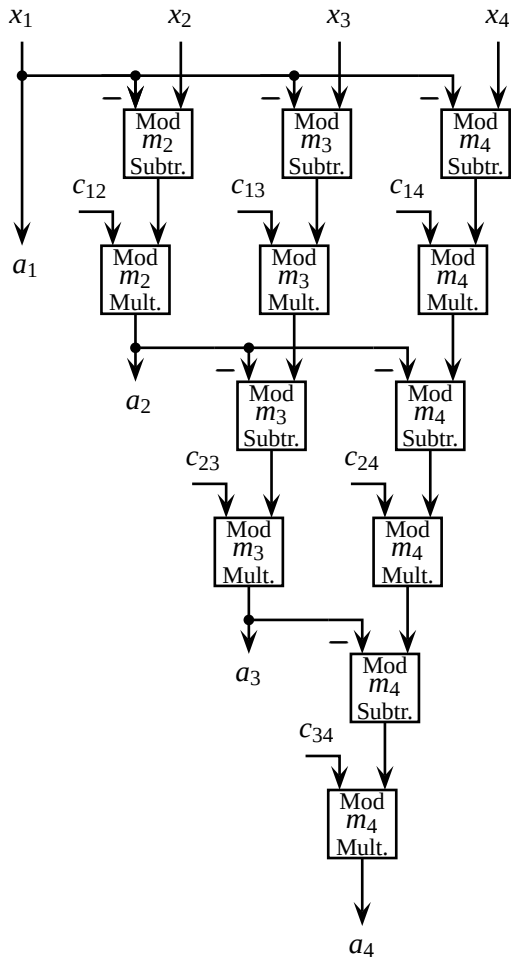

Fig. 1. Mixed-radix algorithm to convert the residue code to a weighted code.

Operands a_1, a_2, a_3 and a_4 are $(2n+1)$-bit, n-bit, $(n+1)$-bit and n-bit, respectively.

4 Hardware Implementation

4.1 MRD to binari convertor

The proposed architecture of RNS to mixed-radix digits conversion is depicted in Figure 2. It contains only modulo subtractors, and one modulo (2^n-1) multiplication by 2^{n-1}. Modulo (2^n-1) multiplication by 2^{n-1} is equivalent to $(n-1)$ bit circular shifting. In our implementation circular shifting $(n-1)$ the left is the same

as one bit right shifting, because mixed-radix digit a_4 has n-bits.

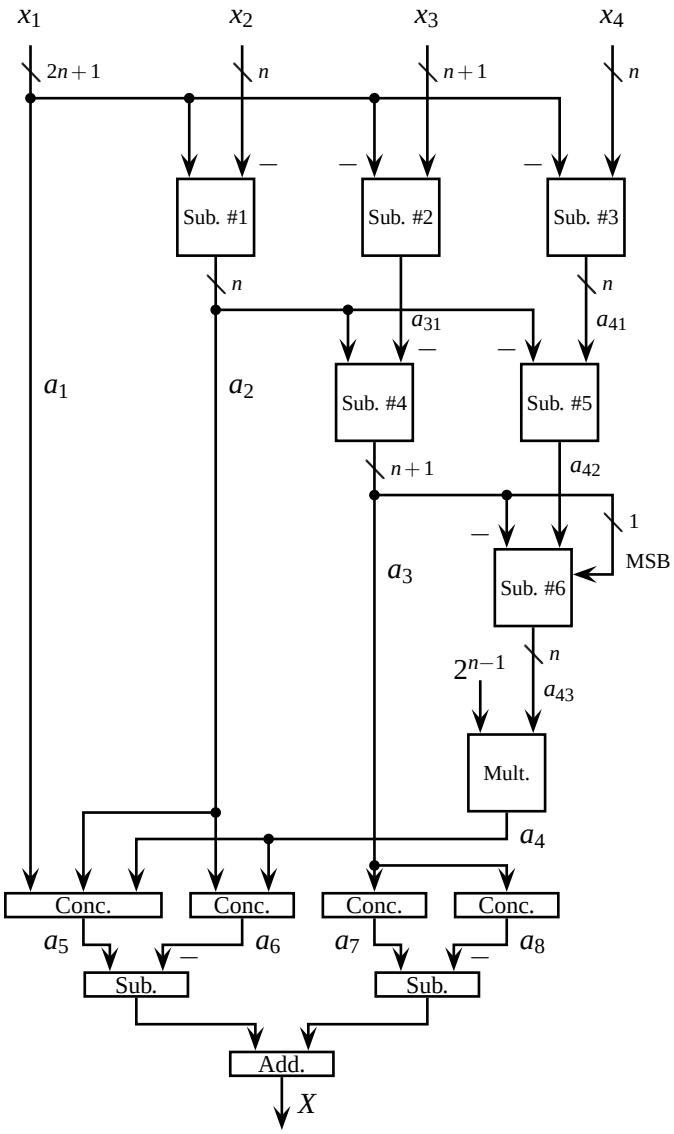

Fig. 2. Mixed-radix convertor for four moduli set $\{2^{2n+1}-1, 2^n, 2^n+1, 2^n-1\}$.

Equation 5 can be simplified as follows

$$\begin{aligned} X &= a_1 + a_2(2^{2n+1}-1) + a_3(2^{2n+1}-1)2^n + a_4(2^{2n-1}+1)2^n(2^n-1) \\ &= a_1 + a_2 2^{2n+1} + a_4 2^{3n+1} + a_4 2^{4n+1} - (a_2 + a_4 2^n + a_4 2^{2n}) + a_3 2^{3n+1} - a_3 2^n \end{aligned} \tag{9}$$

The hardware realization of (9) can be simplified as follows

$$X = a_5 - a_6 + a_7 - a_8 \tag{10}$$

where

$$\begin{aligned}
a_5 =& a_1 + 2^{2n+1}a_2 + 2^{3n+1}a_4 + 2^{4n+1}a_4 \\
=& (\underbrace{a_{1,2n}, a_{1,2n-1}, \ldots, a_{1,0}}_{2n+1}) \\
&+ (\underbrace{a_{2,n-1}, a_{2,n-2}, \ldots, a_{2,0}}_{n}, \underbrace{0,0\ldots,0}_{2n+1}) \\
&+ (\underbrace{a_{4,n-1}, a_{4,n-2}, \ldots, a_{4,0}}_{n}, \underbrace{0,0,\ldots,0}_{3n+1}) \\
&+ (\underbrace{a_{4,n-1}, a_{4,n-2}, \ldots, a_{4,0}}_{n}, \underbrace{0,0,\ldots,0}_{4n+1}) \\
=& (\underbrace{a_{4,n-1}, \ldots, a_{4,0}, a_{4,n-1}, \ldots, a_{4,0}, a_{2,n-1}, \ldots, a_{2,0}, a_{1,2n}, \ldots, a_{1,0}}_{5n+1})
\end{aligned} \tag{11}$$

$$\begin{aligned}
a_6 =& a_2 + a_4 2^n + a_4 2^{2n} \\
=& (\underbrace{a_{2,n-1}, a_{2,n-2}, \ldots, a_{2,0}}_{n}) \\
&+ (\underbrace{a_{4,n-1}, a_{4,n-2}, \ldots, a_{4,0}}_{n}, \underbrace{0,0\ldots,0}_{n}) \\
&+ (\underbrace{a_{4,n-1}, a_{4,n-2}, \ldots, a_{4,0}}_{n}, \underbrace{0,0\ldots,0}_{2n}) \\
=& (\underbrace{0,0,\ldots,0, a_{4,n-1}, \ldots, a_{4,0}, a_{4,n-1}, \ldots, a_{4,0}, a_{2,n-1}, \ldots, a_{2,0}}_{5n+1})
\end{aligned} \tag{12}$$

$$\begin{aligned}
a_7 =& a_3 2^{3n+1} \\
=& (\underbrace{a_{3,n}, a_{3,n-1}, \ldots, a_{3,0}}_{n+1}, \underbrace{0,0\ldots,0}_{3n+1}) \\
=& (\underbrace{0,0,\ldots,0, a_{3,n}, \ldots, a_{3,0}, 0,0,\ldots,0}_{5n+1})
\end{aligned} \tag{13}$$

$$\begin{aligned}
a_8 =& a_3 2^{n} \\
=& (\underbrace{a_{3,n}, a_{3,n-1}, \ldots, a_{3,0}}_{n+1}, \underbrace{0,0\ldots,0}_{n}) \\
=& (\underbrace{0,0,\ldots,0, a_{3,n}, \ldots, a_{3,0}, 0,0,\ldots,0}_{5n+1})
\end{aligned} \tag{14}$$

The following example will demonstrate in detail the procedure of bit organization operands a_5, a_6, a_7 and a_8 for calculating the results of conversion mixed-radix digits to binary number.

Example 1 *Suppose the number* $X = 2084879$ *is given, which is equal to upper limit of the dinamic range. The RNS representation for* $n = 4$*, then* $X \overset{(511,16,17,15)}{\longrightarrow} (510, 15, 16, 14)$*. It is shown, Mixed-radix digit* a_i*,* $i = 1, \ldots, 4$*, have the same digits as the RNS digits.*

a_1	510	111111110
a_2	15	1111
a_3	16	10000
a_4	14	1110

By using bit organization (11), (12), (13) *and* (14) *mixed-radix digits* a_5, a_6, a_7 *and* a_8 *can be evaluated as*

a_5	111011101111111111110
a_6	000000000111011101111
a_7	000100000000000000000
a_8	000000000000100000000

As show in Fig. 2*, in order to calculate the binary number X must be two subtraction and one addition. First binary subtraction* $a_5 - a_6$ *is*

a_5	111011101111111111110
a_6	000000000111011101111
$a_5 - a_6$	111011101000100001111

Second binary subtraction $a_7 - a_8$ *is*

a_7	000100000000000000000
a_8	000000000000100000000
$a_7 - a_8$	000011111111100000000

Finally, the binary number $X_{bin} = (a_5 - a_6) + (a_7 - a_8)$ *is calculated as*

$a_5 - a_6$	111011101000100001111
$a_7 - a_8$	000011111111100000000
Final rezult X_{bin}	111111101000000001111

The following hold true $111111101000000001111_2 = 2084879_{10}$

4.2 RNS to MRD conversion

Binary number x_1 is $2n+1$-bit word and can be written in the following form

$$x_1 = n_2 2^{2n} + n_1 2^n + n_0 \tag{15}$$

were n_0 and n_1 are n-bit words, while n_2 is one bit binary digit.

4.2.1 First subtractor

The value $\langle x_1 \rangle_{2^n}$ can be obtained by remainder of the division of x_1 by 2^n, which can be accomplished by truncating the binary number x_1. Since x_1 is binary number on $2n+1$ bits, then:

$$\langle x_1 \rangle_{2^n} = n_0 \tag{16}$$

The operations modulo 2^n are necessarily "carry-ignore" operations.

4.2.2 Second subtractor

Second subtractor # Sub. 2, shown in Figure 2, contain two parts. First part is convertor $\langle x_1 \rangle_{2^n+1}$ and second part is modulo (2^n+1) subtracter, as is shown in Fig. 3

Since $\langle 2^n c \rangle_{2^n+1} = -c$, calculation $\langle x_1 \rangle_{2^n+1}$ can be performed as a sequence of subtraction and addition, as described below:

$$\langle x_1 \rangle_{2^n+1} = \langle n_2 - n_1 + n_0 \rangle_{2^n+1} \tag{17}$$

the above leads to proposed architecture for the residue of $\langle x_1 \rangle_{2^n+1}$ calculation that is presented in Figure 3(a).

Example 2 *Consider the moduli system* $\{511,16,17,15\}$ *for* $n = 4$ *and given residues* $X = 2006969 = (272,9,0,14)$. *Thus,* $x_1 = 272_{10} = 100010000_2$ *and finally in binary form* $n_0 = 0000$, $n_1 = 0001$ *and* $n_2 = 1$.

Subtractor gives the following result:

n_0	0000	
$-n_1$	0001	
s^1	11111	$s^1 = 1111$ *and* $b_{out} = 1$

First adder (Adder #1) gives the following result:

s^1	1111	
b_{out}	1	
n_2	1	
s^2	10001	$s_n = 1$ *and* $s_0 = 1$; $s_0 \wedge s_1 = 1$

Since $s_0 \wedge s_1 = 1$, *n-bits* 1111 *is added to* s^2 *which yields:*

s^2	10001
n-bits	1111
$\langle 272 \rangle_{17}$	100000 *Carry is 1*

Carry on $n+1$ *position is omitted and convertor returns* $\langle 272 \rangle_{17} = 0$, *which is true.* ■

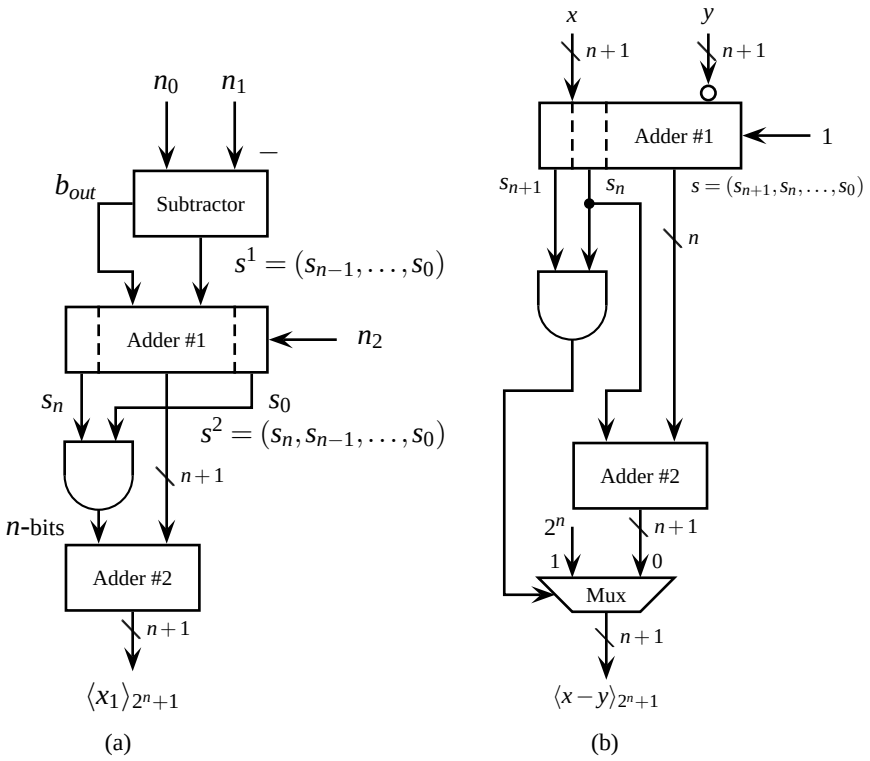

Fig. 3. (a) Convertor $\langle x_1 \rangle_{2^n+1}$. (b) Modulo (2^n+1) subtractor.

The modulo (2^n+1) subtracter, we used architecture shown in Figure 3(b) [3]. The subtraction modulo (2^n+1) can be expressed as follows:

$$\langle x-y \rangle_{2^n+1} = \begin{cases} 2^n & \text{if} \quad x = 2^n \text{ and } y = 0 \\ \langle x + \overline{y} + 1 + s_n \rangle_{2^n} & \text{otherwise} \end{cases} \tag{18}$$

The following example illustrates the modulo (2^n+1) subtraction.

For $x \leq y$ we have $x+\overline{y}+1 \leq 2^{n+1}$ and $s_{n+1}=0$, $s_n=1$. Therefore, using eq. (18) $\langle x-y\rangle_{2^n+1} = x+\overline{y}+1+1$. For $x=00101$ and $y=01111$ by applying Adder #1 we obtain:

x	00101	
$\overline{y}$	10000	
	1	
s	010110	$s_{n+1}=0$ and $s_n=1$; $s_{n+1} \wedge s_n = 0$

The final result, by applying adder #2 for adding last n-bit of previous sum 0110 with $s_n=1$, is 0111. The multiplexer forward this input to the output. This hold true because $\langle -10\rangle_{17} = 7$.

4.2.3 Third subtractor

The third subtractor # Sub. 3, shown in Figure 2, contains two parts. The first part is convertor $\langle x_1\rangle_{2^n-1}$ and the second part is modulo (2^n-1) subtracter, as is shown in Fig. 4

Since $\langle 2^n\rangle_{2^n-1}=1$, calculation $\langle x_1\rangle_{2^n-1}$ can be performed as a sequence of additions and subtractions, as described below:

$$\langle x_1\rangle_{2^n+1} = \langle n_2+n_1+n_0\rangle_{2^n-} \tag{19}$$

the above leads to proposed architecture for the residue of $\langle x_1\rangle_{2^n-1}$ calculation that is presented in Figure 4(a).

Example 3 *Consider* $x_1 = 390_{10} = 1\,1000\,0110_2$. *Adder #1 shown on the Fig. 4(a) perform following result:*

n_0	1000	
n_1	0110	
n_2	1	
s^1	1111	$c_{out1}=0$

Adder #2 perform following result

s^1	1111	
c_{out1}	0	
	1	
s^2	10000	$c_{out2}=1$

The Subtractor gives the final fresult.

s^2	0000
$-\overline{c}_{out2}$	0
$\langle 390 \rangle_{15}$	0000

Thus, result $\langle 390 \rangle_{15} = 0$*, which is true.* ■

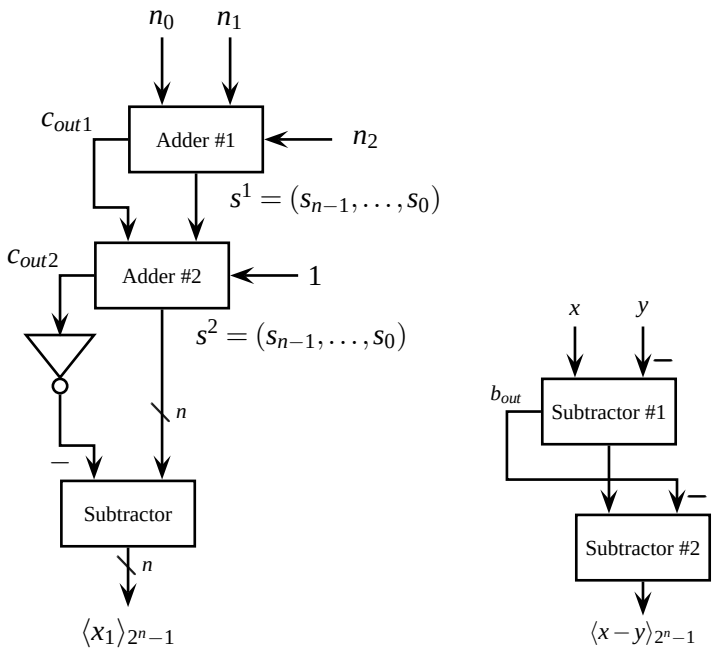

Fig. 4. (a) Convertor $\langle x_1 \rangle_{2^n-1}$. (b) Modulo 2^n-1 subtractor.

The modulo (2^n-1) subtraction can be expressed as follows:

$$\langle x-y \rangle_{2^n-1} = \langle x-y-b_{out} \rangle_{2^n} \tag{20}$$

This type of subtractor is shown on the Fig. 4(b), and it is known as the Borrow-Save Subtractor with End-Around-borrow (BSS with EAB).

4.3 Fourth and fifth subtractor

Subtractor #4, is modulo (2^n+1) subtractor, is shown on the Fig. 3(b), but subtractor #5 is modulo (2^n-1) subtractor (BSS with EAB Fig. 4(b)).

4.3.1 Sixth subtractor

The sixth subtractor # Sub. 6, shown in Figure 2 is very simple as it is shown in Fig. 5. Minuend is n-bit binary number, but subtrahend is $(n+1)$-bit binary number. MSB bit of subtrahend is put on the borrow input of subtractor #1. The second subtractor subtracts borrow back to s^1. That is $\langle x-y\rangle_{2^n-1} = s^1 - b_{out}$; end-around-borrow.

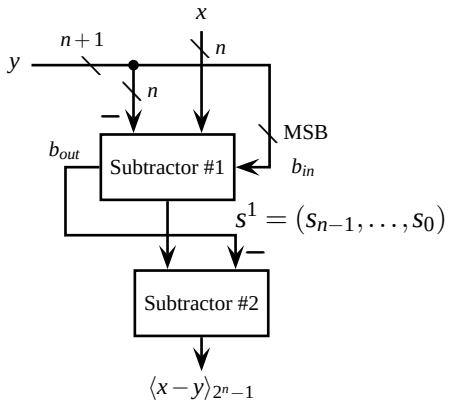

Fig. 5. Oduzimač po modulu $2^n - 1$

5 Conclusion

This paper presents a mixed-radix reverse converter for the recently proposed residue number system moduli set $\{2^n-1, 2^n, 2^n+1, 2^{2n+1}-1\}$. The implementation consists of two levels. The first level is the algorithm to convert RNS number to mixed-radix digits. The algorithm is improved by using optimal choice of form of moduli set. The second level is a hardware architecture. Carry-Save-Adder with End-Around-Carry is replaced with Borrow-Save-Subtractor that avoids two complement operations, and End-Around-Carry adder. Further, the binary subtraction is optimized by using Borrow-Propagate-Subtractor with End-Around-Borrow which avoids one complement operation and multiplexer. The proposed converter architecture is memoryless and it can be efficiently implemented.

Acknowledgement

This paper is supported by the Project Grant III44004 (2011-2014) financed by the Ministry of Education and Science, Republic of Serbia.

The authors wish to thank Dr. Vidosav Stojanović Professor at University of Niš, Faculty of Electronic Engineering, for help and useful discussions.

References

[1] H. L. Garner, "The residue number system," *IRE Trans. Electronic Computer*, vol. EC-8, no. Issue 2, pp. 140–147, Jun. 1959.

[2] K. A. Gbolagade, R. Chaves, L. Sousa, and S. D. Cotofana, "Residue-to-binary converters for moduli set $\{2^{2n+1}-1, 2^{2n}, 2^n-1\}$," in *2nd Int. Circuits on Adaptive Science and Technology (ICAST09)*, Accora, Ghana, Dec. 2009, pp. 26–33.

[3] A. A. Ghouwayel, Y. Louët, and J. Palicot, "A reconfigurable butterfly architecture for fourier and fermat transforms," in *4th Karlsruhe Workshop on Software Radios*, Karsruhe, Geramany, Mar. 2006. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00083992/en/

[4] W. K. Jenkins and B. Leon, "The use of residue number systems in the design of finite impulse response digital filters," *IEEE Trans. on Circuits and Systems*, vol. CAS-24, no. 4, pp. 191–201, Apr. 1977.

[5] P. V. A. Mohan and A. B. Premkumar, "RNS-to-binary converters for two four-moduli sets $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ and $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$," *IEEE trans. on Circuits and System-I:Regular Papers*, vol. 54, pp. 1245–1254, Jun. 2007.

[6] A. S. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, and S. Timarchi, "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 Transactions on Circuits and SystemsI: Regular Papers,*, vol. 57, no. 4, pp. 823–835, Apr. 2010.

[7] S. Pontarelliyz, G. Cardarilliy, M. Rey, and A. Salsanoy, "Totally fault tolerant rns based FIR filters," in *14th IEEE International On-Line Testing Symposium, IOLTS'08.*, Jul. 7–9, 2008, pp. 192–194.

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

[9] W. Wang, M. Swamy, M. Ahmad, and Y. Wang, "A study of residue to binary converters for the three-moduli sets," *IEEE Trans. on Circuits and Syst-Fundamental Theory and Applications*, vol. 50, no. 2, pp. 235–245, 2003.

[10] Y. Wang, "Residue-to-binary converters based on new chinese remainder theorems," *IEEE Trans. on Circuits and Syst-II, Analog and Digital Signal Processing*, vol. 47, no. 3.