ISSN (Online): 2230-7893
www.IJCEM.org

Binary to RNS encoder with Modulo 2^n+1 Channel in Diminished-1 Number System

Ivan Krstic[1], Negovan Stamenkovic[2], Milena Petrovic[3] and Vidosav Stojanovic[4]

[1] Faculty of Techical Science, K.Mitrovica, 38220, Serbia

[2,3] Faculty of Natural Science, K.Mitrovica, 38220, Serbia

[4] Faculty of Electronic Engineering, Nis, 18000, Serbia

Abstract

Architecture of binary to residue number system encoder based on the moduli set $\{2^n-1, 2^n, 2^n+1\}$, and architecture of encoder with modulo 2^n+1 channel in the diminished-1 representation instead of the standard modulo 2^n+1 channel are presented. We consider the binary numbers with dynamic range of proposed moduli set which is $2^{3n}-2^n$. Within this dynamic range, $3n$-bit binary number is partitioned into three n-bit parts and converted to standard residue numbers, or converted to two standard residues and a residue in the diminished-1 representation. Our approach enables a unified design for the moduli set adders. The proposed architecture can be utilized in conjunction with any fast binary adder without requiring any extra hardware. The encoder architecture with diminished-1 encoded channel has been mapped on an Xilinx FPGA chip.

Keywords: RNS system, special moduli set, forward encoder, diminished-1 encoded channel, carry save adder, full adder, FPGA.

1. Introduction

A binary-to-residue encoder is an essential building block for a residue number system and as such it should be built with minimum amount of hardware and efficient in terms of speed and power. Conceptually, binary-to-residue encoder involves computation of remainders of input bit stream with respect to each modulus in the RNS moduli set. In other words, the binary-to-residue encoder maps a binary weighted number into a finite ring.
There are three main architectures for forward conversion (residue generator) that can be used to build binary to residue encoder.

The first architecture is based on the use of look-up tables which involves precomputing of all possible values that conversion requires and storing these values in memory. Parahmi [9] proposed ROM lookup tables for binary to residue conversions which are optimized in terms of size and speed. The disadvantage of the lookup table architecture is the size of the lookup table which increases disproportionately with increase of magnitudes of the moduli.

The second architecture involves recursion in the binary-to-residue generation by means of using efficient arithmetic units called processing elements, connected via a data bus [1]. The binary values are decomposed into an array of power-of-two values, the residues of each power-of-two values are computed and summed up with modulo adders. The proposed structure, containing $n/2$ processing elements, is used to generate the residues that correspond to each and every bit in the n bit binary word. A modification of the above method was proposed by Cappocelli and Giancarlo [3]. They proposed partitioning of n bit binary word into n/q groups of q adjacent bits. The residue corresponding to the first bit in a group of bits is stored in ROM and the residue of next power of two in that group is evaluated. This architecture is suitable for pipelined applications. Based on the approach proposed by Caposelli and Giancarlo, S. Piestrak [10] proposed the concept of periodic properties of modular operations in converter designs. According to the periodic property, the input bits can be divided into a number of groups and handled similarly, which greatly simplifies design. Mohan [8] showed that by proper choice of the moduli set, for example special moduli set, the hardware needed for realization of binary to residue converter can be reduced. Premkumar [12] transferred the conversion operation into modular exponentiation and presented an efficient memory-free architecture design.

The partitioning of the input operand based on the periodicity instead of the wordlength of the modulus is also the cause of large variation in the architectural area and time complexities for the residue generators of different moduli. To eliminate these problems in the most

recently published paper Low and Chang [7] have proposed a new approach to the design of highly efficient residue generators for any arbitrary moduli of up to six bits wide.

The third architecture is memoryless and is related to the power-of-two moduli set, such as the special moduli set $\{2^n-1, 2^n, 2^n+1\}$. The use of special moduli sets simplifies the binary to residue conversion algorithms and binary to RNS encoder architectures. Periodicity of residues allows the computation of the residues of the integer X as follows. Binary representation of integer X is partitioned in k blocks each of n-bits. These blocks can be aligned and added in a CSA tree. Pourbigharaz and Yassine [11] presented a binary to residue architecture for the special moduli set. They derived a formula to compute the residue of an integer with respect to the modulus (2^n+1) and replaced the end around carry (EAC) stage with a simple multiplexer. The proposed architecture is free of any modulo adder. A framework for memoryless binary to residue encoder based on periodic properties of the special modulo numbers has been introduced in paper [13]. Present decoder is memoryless but includes two extra complex correction network. Bi and Jones [2] performed residue to binary encoder based on special moduli set using only eight conventional adders.

Diminished-1 number system [6] is an alternative representation for modulo 2^n+1, in which the remainder of division is represented as $x_3 = \langle X-1 \rangle_{2^n+1}$. Thus, each operand is represented decreased by one, but zero operands are not used in the computation channel. The results of arithmetic operations are derived alternatively when any operand or the result is zero [14, 5]. The diminished-1 representation of modulo 2^n-1 remainder can lead to implementations with delay and area complexity approaching that of the modulo 2^n-1 channel.

This paper presents a new binary-to-residue encoder for the moduli set $\{2^n-1, 2^n, 2^n+1\}$ with $\{2^n+1\}$ diminished-1 modulo channel which use the partitioning method. In our approach, the $3n$-bit input is divided into three n-bit sections in order to obtain the corresponding three residue numbers in parallel. Our approach enables utilization of any design for the moduli set adders. The proposed architecture can be utilized in conjunction with any fast binary adder without involving any extra hardware.

The rest of paper is organized as follows. In Section 2 we introduce binary-to-residue memoryless encoder for special moduli set based only on standard combinational logic. A novel design of binary-to-residue converter with diminished-one $\{2^n+1\}$ encoded channel is presented in Section 3. Section 4 presents the evaluation of performances of both the encoder based on special moduli set and the encoder with diminished-1 encoded channel. Our conclusion is drawn in Section 5.

2. Binary-to-residue encoder

Available architectures for binary-to-residue encoder based on the special three moduli-set $\{2^n-1, 2^n, 2^n+1\}$ are presented here. Dynamic range corresponding to the product of the modulus for this particular moduli set is evidently $M = 2^{3n} - 2^n$ i.e. corresponds to $3n$ bits. Thus, given a $3n$ bit unsigned binary integer X, the residues corresponding to three moduli can be uniquely represented in RNS by a set of residues $X = (x_1, x_2, x_3)$, were x_1 is the reminder when X is divided by modulo 2^n-1 denoted as $\langle X \rangle_{2^n-1}$, $x_2 = \langle X \rangle_{2^n}$ and $x_3 = \langle X \rangle_{2^n+1}$. Based on the definition of RNS, the residue is smaller than its corresponding modulo.

It is well known that an $3n$ bit integer in the range $0 \le X \le M-1$ can be represented in power-of-two notation as [15, 10, 4]:

$$X = \sum_{i=0}^{3n-1} b_i 2^i = N_2 \times 2^{2n} + N_1 \times 2^n + N_0 \qquad (1)$$

where

$$N_0 = \sum_{i=0}^{n-1} b_i \times 2^i, \quad N_1 = \sum_{i=n}^{2n-1} b_i \times 2^{i-n},$$
$$N_2 = \sum_{i=2n}^{3n-1} b_i \times 2^{i-2n} \qquad (2)$$

In order to obtain the RNS representation of the integer X, partitioned into three n-bit parts N_0, N_1 and N_2, three converters are required, one for each channel. The encoder architecture shown in Fig. 1 is based on equations

(3), (6) and (15). The design of each building blocks are briefly described in following subsections.

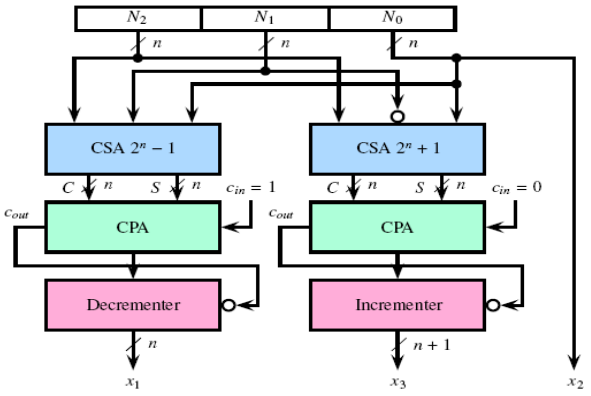

Fig.1Binary-to-residue encoder for special moduli set $\{2^n\text{-}1, 2^n; 2^n + 1\}$.

2.1 Modulo 2^n Channel

The converter for the 2^n channel is very simple. The value x_2 is obtained as remainder of the division of X by 2^n, which can be accomplished by truncating the value X as follows:

$$x_2 = \langle X \rangle_{2^n} = b_{n-1} b_{n-2} \cdots b_0 \tag{3}$$

2.2 Modulo 2^n -1 Channel

For a $2^n - 1$ channel the calculation of a corresponding residue is more complex, since the final result of the conversion depends on the values of all N_i bits. Instead of using a division operation to calculate the $2^n - 1$ residue, which is a complex operation and expensive both in terms of area and speed, the calculation can be performed by a sequence of additions, as described below:

$$x_1 = \langle X \rangle_{2^n-1} = \langle N_2 2^{2n} + N_1 2^n + N_0 \rangle_{2^n-1} \tag{4}$$

By taking the equation:

$$\langle 2^n \rangle_{2^n-1} = 1 \tag{5}$$

equation (4) can be rewritten as:

$$x_1 = \langle N_2 + N_1 + N_0 \rangle_{2^n-1} \tag{6}$$

Thus the conversion of X to modulo $2^n - 1$ can be performed simply by adding components of X (N_i, $i = 1, 2$ and 3) by using carry save adder (CSA) modulo $2^n - 1$ with end around carry (EAC) which generates two n -bit results, n -bit partial sum vector $S = s_{n-1} s_{n-2} \ldots s_0$ and n -bit carry vector $C = c_n c_{n-1} \ldots c_1$. As known, end-around operation is achieved by connecting c_n back to the carry vector as the LSB c_0.

Modulo $2^n - 1$ addition algorithm, that avoids double representation of zero, is defined by simple equation:

$$x_1 = \langle S + C \rangle_{2^n-1} = \langle S + C + 1 - \bar{c}_{out} \rangle_{2^n} \tag{7}$$

where $\bar{c}_{out}$ is the one's complement of carry out which is generated by the carry propagate adder (CPA).

The binary to modulo $2^n - 1$ converter (6) results in a hardware structure presented in Figure 2. The decrementer is implemented as a linear array of a half-subtractors. The half subtractor has the same circuitry as the half adder except for an extra inverter, since borrow and carry are inverse operations.

The theoretical formula of the propagation delay of binary to modulo $2^n - 1$ converter, according to unit gate delay model, comprises the delay from CSA with end around carry (from $N_{0,0}$ to s_0, T_{csa_eac}), CPA with end around inverted carry (from s_0 to c_{out}, T_{cpa_eaic}) and decrementer (from c_{out} to $x_{1,5}$, T_{dec}):

$$\begin{aligned} T_1 &= T_{csa\_eac} + T_{cpa\_eaic} + T_{dec} = 4 + 2n + (n+1) \\ &= 3n + 5 \end{aligned} \tag{8}$$

The area cost of binary to modulo $2^n - 1$ converter is:

$$\begin{aligned} A_1 &= A_{csa\_eac} + A_{cpa\_eaic} + A_{dec} = 7n + 7n + 3n \\ &= 17n \end{aligned} \tag{9}$$

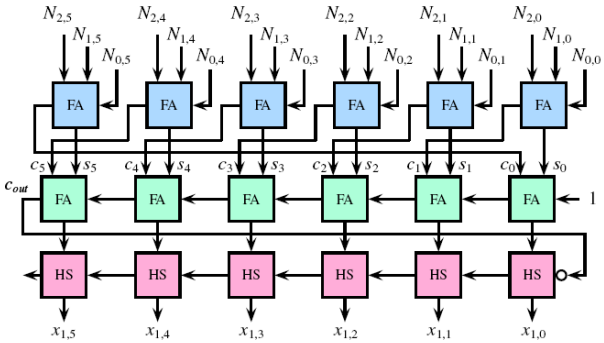

Fig. 2 Binary to modulo $(2^6 - 1)$ converter.

ISSN (Online): 2230-7893
www.IJCEM.org

The validity of the algorithm described above for modulo (2^n-1) binary to RNS converter of 16-th bit number and $n=6$ is demonstrated in the following example. Let $X = 54\,425 = 1101\,010010\,011001$. Then the carry save adder with end around carry provides sum and carry bits.

N_2	=				1	1	0	1
N_1	=		0	1	0	0	1	0
N_0	=		0	1	1	0	0	1
S	=		0	0	0	1	1	0
C	=	0	1	1	0	0	1	
		→						0

The carry propagate adder with end around inverted carry, that decrements the sum of $S+C$, provides the first residue x_1.

S	=		0	0	0	1	1	0
C	=		1	1	0	0	1	0
								1
	=	0	1	1	1	0	0	1
$-\bar{c}_{out}$		→						1
x_1	=		1	1	1	0	0	0

Thus $x_1 = 56$, which can be verified to be true: $\langle 54\,425\rangle_{63} = 56$.

2.3 Modulo 2^n+1 Channel

The 2^n+1 residue can be calculated in an identical manner as residue for the 2^n-1 channel:

$$x_3 = \langle X\rangle_{2^n+1} = \langle N_2 2^{2n} + N_1 2^n + N_0\rangle_{2^n+1} \quad (10)$$

Since

$$\langle 2^n\rangle_{2^n+1} = \langle (2^n+1)-1\rangle_{2^n+1} = \langle -1\rangle_{2^n+1} \quad (11)$$

and

$$\langle 2^{2n}\rangle_{2^n+1} = \langle 2^n \times 2^n\rangle_{2^n+1} = \langle (-1)\times(-1)\rangle_{2^n+1} = \langle 1\rangle_{2^n+1} \quad (12)$$

equation (10) can be rewritten in following form:

$$x_3 = \langle N_2 - N_1 + N_0\rangle_{2^n+1} \quad (13)$$

Further, the negative number can be represented using the 2^n+1 complement system:

$$\langle -N_1\rangle_{2^n+1} = \langle (2^n+1)-N_1\rangle_{2^n+1} = \langle (2^n-1-N_1)+2\rangle_{2^n+1} = \langle \bar{N}_1+2\rangle_{2^n+1} \quad (14)$$

where $\bar{N}_1 = 2^n-1-N_1$ is the one's complement of binary number N_1. Thus equation (13) can be rewritten as:

$$x_3 = \langle\langle N_2 + \bar{N}_1 + N_0 + 1\rangle_{2^n+1} + 1\rangle_{2^n+1} \quad (15)$$

These three additions can be performed by means of using modulo 2^n+1 carry save adder (CSA) with end-around inverted carry (EAIC). Modulo 2^n+1 CSA with EAIC operation is achieved by connecting $\bar{c}_n$ back to the carry vector as the LSB c_0. Fortunately, modulo 2^n+1 carry save adder not only adds the input values, but also intrinsically adds one extra one [4]. Since $S<2^n$, $C<2^n$ and $S+C<2^{n+1}$ are always true, modulo 2^n+1 addition algorithm is defined as proposed in [3]:

$$x_3 = \langle S+C+1\rangle_{2^n+1} = \langle S+C\rangle_{2^n} + \bar{c}_{out} \quad (16)$$

where $\bar{c}_{out}$ is the one's complement of carry out which is generated by the carry propagate adder (CPA).

The binary to modulo 2^n+1 converter, depicted in Figure 3, is very similar to the one depicted in Figure 2.

The theoretical formula of the propagation delay of binary to modulo 2^n+1 converter comprises the delay from CSA with end around inverted carry (from $N_{0,0}$ to s_0, T_{csa_eaic}), CPA with end around inverted carry (from s_0 to c_{out}, T_{cpa_eaic}) and incrementer (from c_{out} to $x_{3,5}$, T_{inc}):

$$T_3 = T_{csa\_eaic} + T_{cpa\_eaic} + T_{inc} = 4 + 2n + (n+1) = 3n+5 \quad (17)$$

The area cost of binary to modulo 2^n+1 converter is:

$$A_3 = A_{csa\_eaic} + A_{cpa\_eaic} + A_{dec} = 7n + 7n + 3n = 17n \quad (18)$$

ISSN (Online): 2230-7893
www.IJCEM.org

Validity of above modulo (2^n+1) binary to RNS converter of 16-th bit number and $n=6$ is demonstrated on following example. Let $X = 54\,425 = 1101\,010010\,011001$. Then the carry save adder with end around inverted carry provides sum and carry bits.

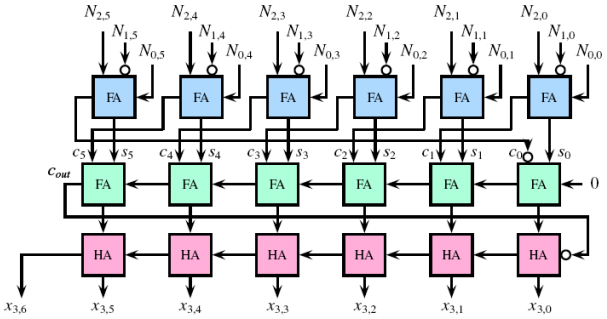

Fig. 3 Binary to modulo (2^6+1) converter.

N_2	=				1	1	0	1
$\overline{N}_1$	=		1	0	1	1	0	1
N_0	=		0	1	1	0	0	1
S	=		1	1	1	0	0	1
C	=	0	0	1	1	0	1	
		→						1

The carry propagate adder with end around inverted carry provides the third residue x_3.

S	=		1	1	1	0	0	1
C	=		0	1	1	0	1	1
	=	1	0	1	0	1	0	0
$+\overline{c}_{out}$		→						0
x_3	=		0	1	0	1	0	0

Thus $x_3 = 20$, which can be verified to be true: $\langle 54\,425\rangle_{65} = 20$.

As shown in discussion above, residue-to-binary encoder can be implemented only by using standard combination logic, full and half adders.

3. Modulo 2^n+1 Channel in the diminished-1 representation

In order to speed up arithmetic operations in the the modulo 2^n+1 channel, the diminished-1 representation of binary numbers introduced in [6] is used. In this representation a number $x \in [0, 2^n]$ is represented as $x' = x-1$, while zero is handled separately.

Since $x_3' = x_3 - 1$ equation (16) can be adopted to compute diminished-1 number x_3' as:

$$x_3' = \langle S + C \rangle_{2^n+1} \tag{19}$$

Residue addition of two n -bit operands $S = s_{n-1}s_{n-2}\ldots s_0$ and $C = c_{n-1}c_{n-2}\ldots c_0$ in modulo 2^n+1 is based on the following relation:

$$x_3' = \langle S + C \rangle_{2^n+1} = \begin{cases} S + C, & \text{if } S + C \le 2^n \\ S + C - (2^n+1), & \text{otherwise} \end{cases} \tag{20}$$

Simple modification of equation (20) gives:

$$x_3' = \langle S + C \rangle_{2^n+1} = \begin{cases} S + C, & \text{if } S + C \le 2^n \\ S + C + (2^n-1) - 2^{n+1}, & \text{otherwise} \end{cases} \tag{21}$$

The carry out signal (c_{out}) resulting from addition of S and C can be used in the process of computing modulo 2^n+1 addition. The carry output is one when $S + C \ge 2^n$.

In order to implement (21) we can ignore the output carry from position 2^{n+1} and add constant value of 2^n-1 to the result of $A = S + C$ if $S + C \ge 2^n$. Considering modulo 2^n+1 addition, the constant value 2^n-1 is $(n+1)$-bit binary number $K = 0\underbrace{11\ldots1}_{n}$, and therefore $B = a_n a_{n-1}\ldots a_0 + 0\underbrace{11\ldots1}_{n}$, where $a_n = c_{out}$. Computation of the x_3' is implemented using a multiplexer that selects either $A = a_n a_{n-1}\ldots a_0$ or $B = b_n b_{n-1}\ldots b_0$ according to value sel. Considering

ISSN (Online): 2230-7893
www.IJCEM.org

the values of c_{out} and b_n, there are three cases to be discussed:

1. If $S+C<2^n$, that is $c_{out}=0$ and b_n can be zero or one, the control signal of the multiplexer should be $sel=0$, connecting the binary number A to the output.
2. If $S+C=2^n$, that is $c_{out}=1$ and $b_n=1$, the control signal of the multiplexer should be also $sel=0$.
3. If $S+C>2^n$, that is $c_{out}=1$ and $b_n=0$, control signal of the multiplexer should be $sel=1$, connectiong binary number B to the output.

According to the above discussion we conclude that $sel=c_{out}\wedge\bar{b}_n$, where $\wedge$ corresponds to the logical AND operation.

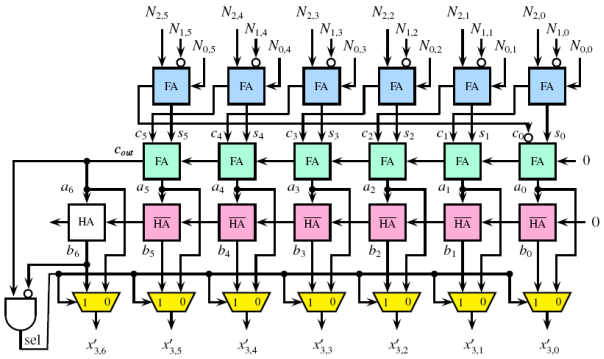

Fig. 4 Binary to modulo (2^6+1) converter in the diminished-1 representation.

Finally, the architecture of our novel binary-to-residue converter with modulo 2^n+1 channel in the diminished-one representation for $n=6$ is given in Fig. 4. The multiplexers are controlled by a sel bit which is produced by the AND gate whose inputs are the carry bit of CPA, c_{out}, and complemented output b_n of a half adder.

The $\overline{\text{HA}}$ in the Fig. 4 is a Full Adder with one input always driven by logical one. Since standard full adder equations are:

$$S=A\,B\,C_{in}+A\,\overline{B}\,\overline{C}_{in}+\overline{A}\,\overline{B}\,C_{in}+\overline{A}\,B\,\overline{C}_{in}$$
$$C_{out}=A\,B+A\,C_{in}+B\,C_{in} \quad (22)$$

if $B=1$ we have

$$S=A\,C_{in}+\overline{A}\,\overline{C}_{in}=\overline{A\oplus C_{in}}$$
$$C_{out}=A+C_{in} \quad (23)$$

where $\overline{A\oplus C_{in}}$ represents exclusive-NOR binary operation. The logic circuit of $\overline{HA}$ is shown in Fig. 5. As can be seen, $\overline{HA}$ has the same complexity as standard half adder except for an extra inverter.

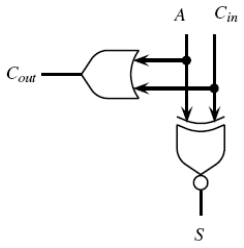

Fig. 5 The logic circuits of a full adder $\overline{\text{HA}}$ whose input *B* is always driven by logical one.

Since constant K in its (n+1)-bit binary representation has the zero at MSB position and the ones in the other bit positions, the combinational network, that produce the sum $A+K$, has standard half adder only at MSB position.

The theoretical formula of the propagation delay of binary to modulo 2^n+1 converter in the diminished-1 representation comprises the delay from CSA with end around inverted carry (from $N_{0,0}$ to s_0, T_{csa_eaic}), CPA with end around inverted carry (from s_0 to a_5, T_{cpa_eaic}), $\overline{\text{HA}}$ (from a_5 to c_{out_ha5}, $T_{\overline{\text{ha}}}$), HA (from c_{out_ha5} to b_6, T_{ha}), AND gate (T_{and}), and multiplexer (from sel to $x'_{3,6}$, T_{mux}):

$$T_{3,D-1}=T_{csa\_eaic}+T_{cpa\_eaic}+T_{\overline{\text{ha}}}+T_{ha}+T_{and}+T_{mux}$$
$$=4+(2n+2)+1+2+1+2=2n+12 \quad (24)$$

The area cost of binary to modulo 2^n+1 converter in the diminished-1 representation is:

$$A_{3,D-1}=A_{csa\_eaic}+A_{cpa\_eaic}+A_{\overline{\text{ha}}_{array}}+A_{ha}$$
$$+A_{and}+A_{mux\_array}=7n+7n+3n+3+1$$
$$+3(n+1)=20n+7 \quad (25)$$

Validity of the modulo (2^n+1) binary to RNS converter in diminished-1 representation for 16-th bit number and $n=6$ is demonstrated in the following example. Let

$X = 54\,425 = 1101\,010010\,011001$. Then the carry save adder with end around inverted carry reduces the three n-bit inputs n_0, n_1 and n_2 to two n-bit number: partial sum (S) and partial carry (C).

N_2	=				1	1	0	1
$\overline{N_1}$	=		1	0	1	1	0	1
N_0	=		0	1	1	0	0	1
S	=		1	1	1	0	0	1
C	=	0	0	1	1	0	1	
		↳						→ 1

The carry propagate adder carry gives

S	=		1	1	1	0	0	1
C	=		0	1	1	0	1	1
A	=	**1**	0	1	0	1	0	0

A carry out $c_{out} = a_6 = 1$ is generated. At last, the two values A and $2^6 - 1$ are then processed using a carry propagate adder based on $\overline{\mathrm{HA}}$. The output is $(n+1)$-bit binary binary number B and carry, which is ignored. Hence,

A	=		**1**	0	1	0	1	0	0
$+(2^n - 1)$	=		0	1	1	1	1	1	1
B	=	1	**0**	0	1	0	0	1	1

where $a_6 = 1$ and $b_6 = 0$. The n-bit multiplexer controlled by sel signal selects the correct sum $x'_{3,6}, \ldots, x'_{3,0}$ from $a_6, \ldots, a_0$ and $b_6, \ldots, b_0$. Since $sel = a_6 \wedge \bar{b}_6 = 1$ the multiplexer selects the outputs of the half adders $b_6 b_5 \ldots b_0$ to yield $x'_3 = 0010011 = 19$ and it can be verified to be true because $\langle 54425 - 1 \rangle_{65} = 19$ is.

4. Simulation Results and Discussion

In this section, propagation delay and the amount of hardware needed for implementation of two binary to residue encoders on ASIC and FPGA are given. The $(2^n + 1)$ modulo channel and $(2^n + 1)$ modulo channel in diminished one number system are alternatively used to design binary-to-residue encoder.

Since residues are computed in paralel, time delays of binary to residue encoders are:

$$T = max(T_1, T_3) = 3n + 5$$
$$T_{D-1} = max(T_1, T_{3,D-1}) = max(3n + 5, 2n + 12) \quad (26)$$

That is, if $n < 7$, $T_{D-1} = 2n + 12$, else $T_{D-1} = 3n + 5$.

Area costs of binary to residue encoders are:

$$A = A_1 + A_3 = 34n$$
$$A_{D-1} = A_1 + A_{3,D-1} = 37n + 7 \quad (27)$$

The algorithms presented in Sections 2 and 3 were used for description of proposed binary to residue encoders in VHDL hardware programming language. Complete design was implemented on Virtex 6 XC6VCX75T FPGA chip using Xilinx ISE Design Suite 14.2 in order to evaluate the cost of physical implementation in terms of used slices and performance in terms of maximum combinational path delay. Behavioral and post-route simulation of implemented encoder were performed using ISIM simulator, which comes as an integral part of Xilinx ISE Design Suite. Exact values which relate to the number of occupied slices and maximum combinational delay (latency) time for different values of n, are shown in Table 1.

Table 1: Hardware utilization of the binary-to-residue encoder

		Encoder			
		Special moduli set		With D-1 channel	
		Latency [ns]	Area [Slice]	Latency [ns]	Area [Slice]
n	4-bit	4.072	28	4.034	29
	5-bit	4.962	36	4.806	38
	6-bit	6.238	45	6.099	47
	7-bit	7.012	52	8.049	54
	8-bit	4.463	74	4.567	77
	10-bit	5.535	90	5.535	95

The VHDL simulation waveforms of the binary to residue encoder based on moduli set $\{63,64,65\}$ with diminished-1 encoded channel are shown in Fig. 6. The length of the input string X is 18 bits and it is sub grouped into three group of 6 bits. The output residues lengths are 6 bits for 2^n-1 and 2^n channels, and 7 bits for 2^n+1 channel.

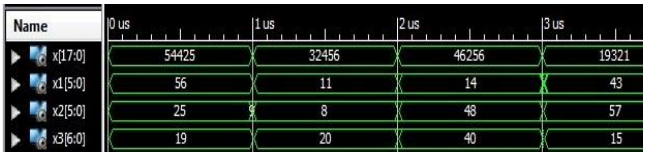

Figure 6: The VHDL simulation waveforms of the binary to residue encoder based on moduli set $\{63,64,65\}$ with diminished-1 encoded channel.

5. Conclusion

In this paper, we investigate binary-to-residue memoryless encoder, which is an important issue concerning the utilization of RNS number system in DSP application. First we demonstrate a binary-to-residue encoder based on the special moduli set $\{2^n-1,2^n,2^n+1\}$. Next, we propose a modulo (2^n+1) generator in diminished-one number system which can be used instead of modulo (2^n+1) channel in binary-to-residue encoder. Additionally, the binary-to-residue encoder with modulo (2^n+1) channel in diminished-1 number system can be easily derived by straightforward modifications of the encoder based on special moduli set with minor hardware overhead.

The obtained results of the design in terms of the number of Xilinx FPGA logic elements and input-to-output propagation delays are given. The speed of the binary to residue encoder can be increased using pipelining. The encoders proposed in this paper are also applicable for ASICs and FPGAs from other vendors.

Acknowledgement

This work was supported by the Serbian Ministry of Science and Technological Development, Project No. 32009TR.

References

[1] B. Parhami, Optimal table-lookup schemes for binary-to-residue and residue-to-binary conversions, in: Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on, Vol. 1, 1993, pp. 812–816.

[2] G. Alia, E. Martinelli, A VLSI algorithm for direct and reverse conversion from weighted binary number system to residue number system, Circuits and Systems, IEEE Transactions on 31 (12) (1984) 1033–1039. doi:10.1109/TCS.1984.1085465.

[3] R. Capocelli, R. Giancarlo, E_cient vlsi networks for converting an integer from binary system to residue number system and vice versa, Circuits and Systems, IEEE Transactions on 35 (11) (1988) 1425–1430. doi:10.1109/31.14466.

[4] S. J. Piestrak, Design of residue generators and multioperand modular adders using carry-save adders, IEEE Transactions on Computers 423 (1) (1994) 68–77.

[5] P. V. A. Mohan, Novel design for binary to rns converters, in: Circuits and Systems, 1994. ISCAS '94., 1994 IEEE International Symposium on, Vol. 2, 1994, pp. 357–360. doi:10.1109/ISCAS.1994.408978.

[6] A. Premkumar, E. L. Ang, E. Lai, Improved memoryless rns forward converter based on the periodicity of residues, Circuits and Systems II: Express Briefs, IEEE Transactions on 53 (2) (2006) 133–137. doi:10.1109/TCSII.2005.857090.

[7] J. Low, C.-H. Chang, A new approach to the design of e_cient residue generators for arbitrary moduli, Circuits and Systems I: Regular Papers, IEEE Transactions on 60 (9) (2013) 2366–2374. doi:10.1109/TCSI.2013.2246211.

[8] F. Pourbigharaz, H. M. Yassine, Modulo-free architecture for binary to residue transformation with respect to $\{2^m-1; 2^m; 2^m+1\}$ moduli set, in: Circuits and Systems, 1994. ISCAS '94., 1994 IEEE International Symposium on, Vol. 2, 1994, pp. 317–320 vol.2. doi:10.1109/ISCAS.1994.408968.

[9] M.-H. Sheu, S.-H. Lin, Y.-T. Chen, Y.-C. Chang, High-speed and reduced-area rns forward converter based on (2^n -1, 2^n, 2^n +1) moduli set, in: Circuits and Systems, 2004. Proceedings. The 2004 IEEE Asia-Pacific Conference on, Vol. 2, 2004, pp. 821–824. doi:10.1109/APCCAS.2004.1413005.

[10] G. Bi, E. V. Jones, Fast conversion between binary and residue numbers, Electronics Letters 24 (19) (1988) 195–197.

[11] L. M. Leibowitz, A simplified binary arithmetic for the Fermat number transform, IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-24 (5) (1976) 356–359.

[12] H. Vergos, C. Efstathiou, A unifying approach for weighted and diminished-1 modulo 2*n* + 1 addition, Circuits and Systems II: Express Briefs, IEEE Transactions on 55 (10) (2008) 1041–1045. doi:10.1109/TCSII.2008.2001964.

[13] C. Efstathiou, I. Voyiatzis, N. Sklavos, On the modulo 2n+1 multiplication for diminished-1 operands, in: Signals, Circuits and Systems, 2008. SCS 2008. 2nd International Conference on, Monastir, Tunisia, 2008, pp. 1–5. doi:10.1109/ICSCS.2008.4746907.

[14] B. Vinnakota, V. V. B. Rao, Fast conversion techniques for binary-residue number systems, IEEE Trans. On Circuits And Systems-I: Fundamental Theories And Applications 41 (12) (1994) 927–929.

[15] R. Chaves, L. Sousa, {2n + 1; 2n+k; 2n − 1}: A new RNS moduli set extension, in: Proceedings of the EUROMICRO Systems on Digital System Design (DSD'04), Remes, France, 2004, pp. 210–217.
[16] S.B. Gaikwad, "Frachtional Fourier Transform of Tempered Boehminans", International Journal of Computational Engineering & Management IJCEM, Vol. 15 Issue 4, July 2012.

Ivan Krstić received the B.Sc. and M.Sc. degrees from the Department of Electronics and Telecommunication at the Faculty of Technical Sciences, Unversity of Priština, Kosovska Mitrovica, in 2011. and 2012. From 2013, he is working as a research assistant at Faculty of Technical Sciences, University of Pristina. He is a PhD student on Faculty of Electronic Engineering, University of Niš, Serbia.

Negovan Stamenković received the M.Sc. degree from the Department of Electronics and Telecommunication at the Faculty of Technical Sciences, University of Priština, Kosovska Mitrovica in 2006 and the Ph.D. degree in electrical and computer engineering from the Faculty of Electronic Engineering, Niš, Serbia, in 2011. He is assistant professor at Faculty of Natural Sciences, University or Pristina. His research is based on signal processing residue number system.

Milena Petrović graduated at Faculty of Mathematics, Belgrade University in 2001 and at Lund University, Department of Numerical Analysis at 2006. She worked as a teacher of mathematics in primary school till 2009. From 2009, she is working as a research assistant at Faculty of Mathematics, University of Pristina. She is a PhD student on Mathematical Faculty, University of Nis, Serbia.

Vidosav S. Stojanović received the B.Sc. degree at the Faculty of Electronic Engineering, University of Niš, in 1964, M.Sc. degree at the Faculty of Electrical Engineering, University of Belgrade, in 1973. and PhD degree at the Faculty of Electronic Engineering, University of Niš, in 1976. After graduating, he worked as research assitant at the Faculty of Electronic Engineering, University of Niš. As a scholar of Houmboldt foundation he spent two years (from 1981 to 1983) at Institute of Telecommunication, University of Munich, Federal Republic of Germany, where he was working on the design of a digital high bit speed transmission system. From 1984 to 1989 he was director of the Institute for Research and Development of Electronic Industry, Niš, and adjunct professor of digital image processing at Faculty of Electronic Engineering, Niš. After five years in the industry he was elected full member professor for Analog Electronics and Digital Signal Processing at Department of Electronic at Faculty of Electronic Engineering, University of Niš. He also was teaching at the University of Priština (Faculty of Electrical Engineering and Faculty of Natural Sciences) and Universities of Podgorica and Sarajevo (Faculty of Electrical Engineering), He was editor in chief of international scientific journal: Facta Universitatis, Series: Electroincs and Energetics from 1993 to 2012. His main research interest is design of analog and digital signal processing systems. He has authored and co-authored four books and over 100 scientific papers published in scietific journals of world famous publishers or referral at scientific conferences.