BINARITY TO RNS ENCODER FOR THE MODULI SET 
\{2^{n-1}, 2^n, 2^n + 1\} WITH EMBEDDED DIMINISHED-1 CHANNEL
FOR DSP APPLICATION

Ivan Krstić, Negovan Stamenković, Vidosav Stojanović

1University of Priština, Faculty of Technical Science, Serbia
2University of Priština, Faculty of Natural Sciences and Mathematics, Serbia
3University of Niš, Faculty of Electronic Engineering, Serbia

Abstract: A binary-to-residues encoder (forward encoder) is an essential building block for the residue number system digital signal processing (RNS DSP) and as such it should be built with a minimal amount of hardware and be efficient in terms of speed and power. The main parts of the forward encoder are residue generators which are usually classified into two categories: the one based on arbitrary moduli-set which make use of look-up tables, and the other based on the special moduli sets. A new memoryless architecture of binary-to-RNS encoder based on the special moduli set \{2^{n-1}, 2^n, 2^n + 1\} with embedded modulo 2^n + 1 channel in the diminished-1 representation is presented. Any of two channels (standard modulo 2^n + 1, or modulo 2^n + 1 in the diminished-1 representation) operation can be performed by using a single switch. The proposed encoder has been implemented on a Xilinx FPGA chip for the various dynamic range requirements.

Keywords: RNS system, special moduli set, forward encoder, diminished-1 encoded channel, modulo carry save adders, Virtex FPGA.

1 Introduction

Residue Number System [1, 2] is a non-weighted integer number system in which arithmetic operations are limited to the addition, subtraction and multiplication. Other arithmetic operations such as division, sign detection, overflow, scaling
and magnitude comparison are non-modular and quite complex for implementation. The RNS is defined in terms of a set of relatively prime moduli called RNS basis.

Special moduli set \{2^n - 1, 2^n, 2^n + 1\} has gained popularity and is expected to play an important role in RNS digital signal processing [3]. In comparison with the other moduli sets, special moduli set has the advantage of low-cost forward conversion, modulo reduction, and the reverse conversion. Thus, the use of this moduli set can significantly reduce hardware complexity and delay [4].

The RNS DSP consists of three major parts: binary-to-residue encoder, modular arithmetic channels and residue to binary decoder (reverse converter). The forward encoder and reverse converter are needed to achieve the RNS representation of the binary number and vice versa, respectively [5]. The above-mentioned modular operations, required by each modular arithmetic channel, are inherently carry-free addition, multiplication and borrow-free subtraction, which means that each digit of the resulting number is a function of only one digit from each operand and independent of the others. This is the most attractive feature of RNS that enables one to design highly parallel structures for computation which leads to speed improvement required for the DSP applications [6, 7, 8, 9, 10].

A binary-to-residues encoder is an essential building block for a residue number system and as such it should be built with a minimal amount of the hardware and along with that be efficient in terms of speed and power. Conceptually, binary-to-residues encoder involves computation of the remainders of the input bit stream with respect to the each modulus in the RNS moduli set. In other words, the binary-to-residues encoder maps a binary weighted number into a finite ring [11]. A finite ring is a set of finite elements over which the modular addition and the modular multiplication operations are defined.

Main parts of binary-to-residues encoder are the forward converters (residue generators). The forward converters are usually classified into the next two categories: the one based on the arbitrary moduli-sets [12, 4] which are usually built using the look-up tables, and the other based on special moduli-sets [13, 14, 15]. The use of special moduli-sets simplifies the forward conversion algorithms and such forward converters can be realized using only combinational logic.

The dynamic range of RNS system, which is equal to the product of the modulus of three moduli-set base \{2^n - 1, 2^n, 2^n + 1\}, is \(M = 2^{3n} - 2^n\) i.e. corresponds to the \(3n\) bits. Thus, any \(3n\)-bit unsigned binary integer \(X\) can be uniquely represented by its residues: \(X = (x_1, x_2, x_3)\), where \(x_1\) 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 \(x_3 = \langle X \rangle_{2^n + 1}\).

The diminished-1 number system [16] can be used to represent modulo \(2^n + 1\) residue \((x_3 = \langle X \rangle_{2^n + 1})\) as: \(x'_3 = \langle X - 1 \rangle_{2^n + 1}\). Thus, the each operand is represented decreased by one, and the zero operands are not used in the computation channel.
In the diminished-1 representation, $x'_3$ is represented as $2^n x_{3,n} + X'_3$, where $x_{3,n}$ is zero indication bit, and $X'_3$ is $n$-bit number part. If $x'_3 > 0$, $x_{3,n} = 0$ and $X'_3 = X_3 - 1$, whereas for $x'_3 = 0$; $x_{3,n} = 1$, and $X'_3 = 0$. Thus, for the diminished-1 representation, the residue of $X - 1$ modulo $2^n + 1$ instead of the residue $X$ modulo $2^n + 1$ is used.

The results of arithmetic operations are derived alternatively when any of operands or the result is equal to zero [17, 18, 19]. In this way, the diminished-1 representation can lead to the implementations with delay and area complexity approaching that of the modulo $2^n - 1$ channel.

This paper presents a binary-to-residues encoder based on special moduli set \{2^n - 1, 2^n, 2^n + 1\} with embedded diminished-1 encoded channel, which unifies the encoders architectures presented in [15]. Theoretical background of forward converters for $2^n - 1$ and $2^n$ channel remains the same as in [15], while new forward converter for $2^n + 1$ channel with embedded diminished-1 encoded channel has been developed. The standard and the diminished-1 forward converters for modulo $2^n + 1$ channel are implemented on the same hardware. Thus, the standard $2^n + 1$ channel or the diminished-1 channel can be activated simply, by using the single switch.

The rest of the paper is organized as follows. In Section 2 we introduce the binary-to-residues memoryless encoder for special moduli set based only on the standard combinational logic and a novel design of the binary to residue encoder with embedded diminished-1 channel. Section 3 presents the hardware implementation and performance evaluation. Our conclusion is drawn in Section 4.

2 Binary-to-residues encoder

In our approach, the $3n$-bit input is divided into three $n$-bit sections to obtain the corresponding three residue numbers in parallel. An $3n$-bit integer in the range $0 \leq X \leq M - 1$ can be represented in power-of-two notation as [20, 21]:

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

where

$$N_0 = \sum_{i=0}^{n-1} b_i 2^i, \quad N_1 = \sum_{i=n}^{2n-1} b_i 2^{i-n} \quad \text{and} \quad N_2 = \sum_{i=2n}^{3n-1} b_i 2^{i-2n}.$$  

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 residue generators are required, one for each channel.
2.1 Forward conversion for modulo $2^n$ and modulo $2^n - 1$ channel

Forward conversion for modulo $2^n$ channel is quite simple, i.e. the residue $x_2$ can be obtained by truncation of $X$:

$$x_2 = \langle X \rangle_{2^n} = b_{n-1}b_{n-2} \cdots b_0.$$  \hspace{1cm} (3)

The calculation of $x_1$ can be performed as a sequence of additions [15]:

$$x_1 = \langle N_2 + N_1 + N_0 \rangle_{2^n-1},$$  \hspace{1cm} (4)

which can be performed by CSA with EAC (carry save adder with end around carry) on whose inputs three $n$-bit operands ($N_2, N_1, N_0$) are connected, followed by the CPA with EAIC (carry propagate adder with end around inverted carry) and decremner.

Slight modifications to the architecture of the modulo $2^n - 1$ residue generator presented in [15] are introduced: LSB full-adder of CPA with EAIC (end-around inverted carry) has been replaced with HA (full-adder with one input driven by the logical one), while MSB half-subtractor of decremner has been replaced with XOR gate, Fig. 1. HA has the same complexity as the standard half-adder except for an extra inverter, whose delay and area consumption can be ignored if using the unit-gate model as a means of performance evaluation.

The critical path of the binary-to-modulo $2^n - 1$ converter is depicted by a dashed line.

Propagation delay of binary to modulo $2^n - 1$ converter, according to the unit-gate model, is $T_1 = 3n + 4$. The area cost of binary to modulo $2^n - 1$ converter is
\[ A_1 = 17n - 5. \]

### 2.2 Forward conversion for modulo \(2^n + 1\) channel

Modulo \(2^n + 1\) residue and modulo \(2^n + 1\) residue in diminished-1 number system representation can be calculated by equations derived in [15]:

\[ x_3 = (S + C + 1)_{2^n + 1}, \quad (5) \]
\[ x'_3 = (S + C)_{2^n + 1}, \quad (6) \]

where \(S\) and \(C\) are \(n\)-bit partial sum and carry vectors generated by modulo \(2^n + 1\) carry save adder, whose inputs are driven by \(N_2, N_1\) and \(N_0\), where \(N_1 = 2^n - 1 - N_1\) is the one’s complement of operand \(N_1\) [22].

By introduction of a control bit \(d\)

\[ d = \begin{cases} 1, & \text{if calculating } x_3 \\ 0, & \text{if calculating } x'_3 \end{cases}, \quad (7) \]

equations (5) and (6) are combined to form

\[ x'''_3 = (S + C + d)_{2^n + 1}. \quad (8) \]

The modulo \(2^n + 1\) addition of two \(n\)-bit operands \(S = s_{n-1}s_{n-2}...s_0\), \(C = c_{n-1}c_{n-2}...c_0\) and the 1-bit operand \(d\) is based on the following relation

\[ x'''_3 = \begin{cases} S + C + d, & S + C + d \leq 2^n \\ S + C + d + (2^n - 1) - 2^{n+1}, & \text{otherwise} \end{cases}. \quad (9) \]

In order to implement (9) we can ignore the output carry \((c_{out})\) from \(2^{n+1}\) position and add the constant value of \(2^n - 1\) to the result of \(A = S + C + d\) (in binary notation \(A = a_na_{n-1}...a_0\), if \(S + C + d > 2^n\). That is, the output of the residue generator should yield the value \(B\) which is obtained by adding the \((n + 1)\)-bit binary number \(K = 011...1\) to the binary number \(A\), \(B = a_na_{n-1}...a_0 + 011...1\), where \(a_n = c_{out}\).

Let \(p_{i+1}\) denote the carry from \(i\)-th bit position obtained while performing addition of the binary numbers \(A\) and \(K\). It is obvious that:

\[ p_1 = a_0, \]
\[ p_{i+1} = p_i \lor a_i, \text{ for } i = 1, 2, \ldots, n - 1, \quad (10) \]

where \(\lor\) corresponds to the logical OR operation.
Furthermore, the output vector $B$ is:

\[
\begin{align*}
b_0 &= \overline{a_0}, \\
b_i &= p_i \oplus \overline{a_i}, \quad \text{for } i = 1, 2, \ldots, n - 1, \\
b_n &= p_n \oplus a_n.
\end{align*}
\] (11)

According to the value of the control signal $sel$ either $A = a_n a_{n-1} \ldots a_0$ or $B = b_n b_{n-1} \ldots b_0$ should be connected to the output (if $sel = 0$ then $x'_3 = A$, else $x'_3 = B$):

\[
x''_{3,k} = (b_k \land sel) \lor (a_k \land \overline{sel}).
\] (12)

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

1. If $S + C + d < 2^n$, that is $c_{out} = 0$ and $p_n$ can be zero or one, then $sel = 0$, which corresponds to the binary number $A$ at the output.
2. If $S + C + d = 2^n$, that is $c_{out} = 1$ and $p_n = 0$, then $sel = 0$ and $x''_3 = A$.
3. If $S + C + d > 2^n$, that is $c_{out} = 1$ and $p_n = 1$, then $sel = 1$, which corresponds to the binary number $B$ at the output.

According to the above discussion it can be concluded that $sel = c_{out} \land p_n$, where $\land$ corresponds to the logical AND operation.

Equation (12) can be simplified by putting (11) in (12):

\[
\begin{align*}
x''_{3,0} &= a_0 \oplus sel, \\
x''_{3,i} &= (sel \land \overline{p_i}) \oplus a_i, \quad \text{for } i = 1, 2, \ldots, n - 1, \\
x''_{3,n} &= a_n \land \overline{p_n}.
\end{align*}
\] (13)

Finally, the architecture of the binary to the modulo $2^n + 1$ converter with the embedded modulo $2^n + 1$ channel in the diminished-1 representation for $n = 6$ is given in the Fig. 2. Depending on the value of the control bit $d$, the converter gives either $x_3$ or $x'_3$. The critical path of the converter is depicted by the dashed line.

The theoretical formula for the propagation delay, i.e. conversion time, of the binary to the modulo $2^n + 1$ residue generator with the embedded diminished-1 channel is $T_3 = 2n + 10$. The area cost is equal to $A_3 = 18n$.

The validity of the modulo $2^n + 1$ channel and the diminished-1 encoded channel operation of the binary to residues encoder for the 16-th bit input number and $n = 6$ is demonstrated in the following example. Let $X = 54 425 = 1101 010010 011001$. The carry save adder with end around inverted carry (EAIC) reduces the three 6-bit inputs $N_0$, $N_1$ and $N_2$ to the two 6-bit numbers: the partial sum sequence ($S$) and the partial carry sequence ($C$)
Fig. 2. The architecture of binary to modulo \((2^6 + 1)\) converter with the embedded modulo \((2^6 + 1)\) channel in the diminished-1 representation. For \(d = 1\) we have \(x_3 = (X)_{2^6+1}\), but for \(d = 0\) we have diminished-1 encoded channel \(x_3' = (X - 1)_{2^6+1}\).

\[
\begin{array}{c}
N_2 = 0 0 1 1 0 1 \\
\bar{N}_1 = 1 0 1 1 0 1 \\
N_0 = 0 1 1 0 0 1 \\
\hline
S = 1 1 1 0 0 1 \\
C = 0 0 1 1 0 1 \\
\end{array}
\]

For \(d = 1\) the carry input of LSB full adder of the CPA adder is equal to 1, and the CPA gives:

\[
\begin{array}{c}
S = 1 1 1 0 0 1 \\
C = 0 1 1 0 1 1 \\
d = 1 \\
A = 1 0 1 0 1 0 \\
\end{array}
\]

A carry out \(c_{out} = a_6 = 1\) is generated. Since the 6-bit vector \(p\) is equal to

\[
p = [1 1 1 1 1 1]
\]

and the \(sel = c_{out} \wedge p_6 = 1\) (bit \(p_6\) is MSB), the output of the converter is given by:

\[
x'''_3 = x_3 = [0 0 1 0 1 0 0]
\]
or in the decimal notation $x_3 = 20$, which can be verified as true.

For $d = 0$, the CPA gives:

\[
\begin{align*}
S &= 1 1 1 0 0 1 \\
C &= 0 1 1 0 1 1 \\
A &= 1 0 1 0 1 0 0
\end{align*}
\]

A carry out $c_{out} = a_6 = 1$ is generated. Since the 6-bit vector $p$ is equal to

\[
p = [1 1 1 1 0 0]
\]

and the $sel = c_{out} \land p_6 = 1$, the output of the converter is given by:

\[
x'_3 = x_3 = [0 0 1 0 0 1 1]
\]

or in the decimal notation $x'_3 = 19$, which can be verified as true.

3 Hardware implementation and performance evaluation

In this section, the propagation delay and the amount of the hardware needed for implementation of proposed encoder on an ASIC and FPGA chip, along with the comparison to the encoders presented in [15] are given.

The encoder architecture shown in Fig. 3 is based on equations (3), (4) and (9).

![Fig. 3. The architecture of the new binary to residues encoder for moduli set \{2^n - 1, 2^n, 2^n + 1\} with embedded diminished-1 channel](image)

The carry save adder with the end around carry, which is an adder on whose inputs three $n$-bit operands are connected, followed by the decremener is used for the modulo $2^n - 1$ channel.
By using the switch $s_w$, the modulo $2^n + 1$ channel in the new encoder can yield either the modulo $2^n + 1$ residue ($x_3$) or the modulo $2^n + 1$ residue represented in diminished-1 number system ($x'_3$), as illustrated in Fig. 3. If the carry input of the CPA is equal to 1 ($s_w$ in upper position) the output of modulo ($2^n + 1$) channel is equal to $x_3$. On the other side, if the carry input of the CPA is equal to 0 ($s_w$ in lower position) the output of the modulo $2^n + 1$ channel is equal to the modulo $2^n + 1$ residue in diminished-1 representation denoted as $x'_3$.

As is shown in Figure 3, the CPA is critical element in the both modulo ($2^n - 1$) and modulo ($2^n + 1$) residue generator data paths. The encoder performance can be increased if the CPA is replaced with parallel prefix adder. In the VLSI implementation, the parallel-prefix adders (or carry-tree adders) are known to have the best performance [23]. However, this performance advantage does not translate directly into an FPGA chip due to constraints on the logic block configurations and routing overhead. An FPGA chip, such as the Virtex-6, contains the number of slices, each containing a number of multiplexers, look-up tables, logic gates, flip-flops, etc. The parallel-prefix adder implementation on an FPGA chip is given in [24]. However, the implementations of the binary to residues encoder based on the parallel-prefix adders can lead to the higher hardware cost and consequently considerable the power dissipation in comparison to the carry propagate based architectures.

Since the residues are computed in parallel, the propagation delay of binary to residues encoder is $T = \max(T_1, T_3)$. That is, if $n < 6$, $T = 2n + 10$, else $T = 3n + 4$. The area cost of binary to residues encoder is $A = A_1 + A_3 = 35n - 5$.

The presented algorithms were used for the description of proposed binary to residues encoder in the VHDL hardware programming language. Complete design was implemented on the Virtex 6 XC6VCX75T FPGA chip using Xilinx ISE Design Suite 14.2 while behavioral and post-route simulation of implemented encoder was performed using ISIM simulator.

The exact values of the area cost and propagation delay which relate to the ASIC and the FPGA-based implementations of the presented encoder architecture for different values of $n$, along with the comparison to architectures presented in [15], are shown in Tables 1 and 2.

<table>
<thead>
<tr>
<th>Design</th>
<th>Delay</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder [15]</td>
<td>$3n + 5$</td>
<td>$34n$</td>
</tr>
<tr>
<td>Encoder with D-1 channel [15]</td>
<td>$\max(2n + 12, 3n + 5)$</td>
<td>$37n + 7$</td>
</tr>
<tr>
<td>New encoder</td>
<td>$\max(2n + 10, 3n + 4)$</td>
<td>$35n - 5$</td>
</tr>
</tbody>
</table>

The encoder shown in Fig. 3 can perform the operation of the both encoders.
presented in [15]. As can be seen from Table 1 there is no significant propagation delay improvement if the ASIC implementation is considered. However, the increase of performance regarding area consumption is significant in comparison to the encoder with the modulo $2^n + 1$ residue in the diminished-1 number system representation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4.072 / 28</td>
<td>4.034 / 29</td>
<td>3.885 / 31</td>
</tr>
<tr>
<td>5</td>
<td>4.962 / 36</td>
<td>4.806 / 38</td>
<td>4.529 / 41</td>
</tr>
<tr>
<td>6</td>
<td>6.238 / 45</td>
<td>6.099 / 47</td>
<td>5.370 / 48</td>
</tr>
<tr>
<td>7</td>
<td>7.012 / 52</td>
<td>8.049 / 54</td>
<td>6.500 / 58</td>
</tr>
<tr>
<td>8</td>
<td>4.463 / 74</td>
<td>4.567 / 77</td>
<td>4.618 / 77</td>
</tr>
<tr>
<td>10</td>
<td>5.535 / 90</td>
<td>5.535 / 95</td>
<td>5.563 / 96</td>
</tr>
</tbody>
</table>

Functional simulation waveforms of the binary to residues encoder based on moduli set \{63,64,65\} with diminished-1 encoded channel are shown in Fig. 4. The length of the input string $X$ is 18 bits and it is sub-grouped into three groups of 6 bits. The output residues are 6 bits long for modulo $2^6 - 1$ and modulo $2^6$ channels, and 7 bits long for modulo $2^6 + 1$ channel.

![Fig. 4. Functional simulation waveforms of the binary-to-residue encoder based on moduli set \{63,64,65\} with diminished-1 encoded channel.](image)

### 4 Conclusion

In this paper, we investigated the binary to residues memoryless encoder, which is an important issue concerning the utilization of the RNS number system in DSP application. We proposed a new binary to residues encoder for moduli set \{$2^n - 1, 2^n, 2^n + 1$\} with embedded diminished-1 channel which can be used instead of standard modulo $2^n + 1$ channel. The modulo $2^n + 1$ channel and the diminished-1 channel are implemented on the same hardware and the encoder can be used to perform either the modulo $2^n + 1$ or the diminished-1 channel operation. The single switch is used for the channel selection. Our approach avoids the initial
calculation of $X - 1$ in order to compute the residue of $X$ modulo $2^n + 1$ residue for the diminished-1 encoded channel and enables utilization of any modulo binary adder designs.

The speed of the binary to residues encoder can be increased by using pipelining. The encoder proposed in this paper is applicable to the ASIC and the FPGA implementations. The obtained results of the design in terms of the number of Xilinx FPGA logic elements and input-to-output propagation delays are given.

Acknowledgement

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

References


