An Efficient Reverse Converter for the Three-Moduli Set $(2^{n+1}-1, 2^n, 2^n-1)$

Ahmad Hiasat

***Abstract*—The well-known three-moduli set $(2^n+1, 2^n, 2^n-1)$, where n is a positive integer, has received considerable attention over the last three decades. Many researchers have proposed residue-to-binary (reverse) converters for this moduli set. However, the problem of designing a reverse converter for the moduli set $(2^{n+1}-1, 2^n, 2^n-1)$ did not receive significant attention. In this brief, we are proposing a reverse converter for this particular moduli set. When compared with the available similar converters, considerable reductions have been achieved. Using VLSI design tools, the area, time, and power reductions are in the range of $(10.6-60.1)\%$, $(13.8-25.2)\%$, and $(7.7-52.1)\%$, respectively.**

***Index Terms*—Binary number system, residue number system (RNS), residue to binary converters, reverse converters.**

I. Introduction

RESIDUE number system (RNS) is a nonweighted representation that exhibits a carry-free arithmetic as far as addition, subtraction, and multiplication are concerned. This feature enables computations on each residue digit to be carried independently of other digits. This independence eliminates complex interconnections among different arithmetic circuits and speeds up computations. Therefore, RNS has been used in applications that require high-speed computations and depend on the above mentioned arithmetic operations. These applications include cryptography, number theoretic transforms, and signal processing [2]–[5].

The RNS is represented using a set of L moduli. The dynamic range is determined by the product of all L moduli. The three-moduli set $(2^n+1, 2^n, 2^n-1)$, where n is a positive integer, has received a considerable attention over the last three decades. Researchers have proposed many designs for a residue to binary (reverse) converter for this moduli set. While early designs were very demanding in terms of hardware and time requirements [6], the very last designs were significantly simpler [7].

Other three-moduli sets that have the form $(2^{n\pm1}-1, 2^n, 2^n-1)$ were also proposed, where, two of three moduli have the form (2^a-1) and no moduli of the form (2^a+1) [8]–[10]. The moduli set of the form $(2^{n+1}-1, 2^n, 2^n-1)$ presented in [10], is a vertical extension to $(2^{n-1}-1, 2^n, 2^n-1)$. The latter set was presented in [8] and [9].

Other efforts have been put to extend, horizontally, the moduli sets by increasing L. Researchers have presented converters for moduli sets with four, five or more moduli [11]–[14]. Nevertheless, work on sign detection, for example, in these horizontally extended sets is still limited and very complicated. However, sign detection for three-moduli sets are significantly simpler and more efficient [15]–[16].

The published work on reverse conversion for residue values expressed using moduli sets of the form $(2^{n\pm1}-1, 2^n, 2^n-1)$ has been very limited [8]–[10]. The converters for such moduli set are still demanding in terms of area and delay compared with the popular set $(2^n+1, 2^n, 2^n-1)$. This brief is an additional effort to reduce the area and time conversion requirements of the moduli set $(2^{n+1}-1, 2^n, 2^n-1)$.

The notational convention adopted in this brief is given as follows.

1) The moduli set under consideration is $(m_1, m_2, m_3) = (2^{n+1}-1, 2^n, 2^n-1)$, where the three moduli are pairwise relatively prime positive integers.
2) $M = \prod_{i=1}^{3} m_i$, dynamic range.
3) The RNS representation of X is (R_1, R_2, R_3), where $X \in [0, M)$.
4) $R_i = \langle X \rangle_{m_i}$, the least non-negative remainder when X is divided by the modulus m_i.
5) The binary representation of RNS digits (R_1, R_2, R_3), are:
   a) $R_1 = \sum_{i=0}^{n} r_{1i}2^i = r_{1n} \cdots r_{11}r_{10}$;
   b) $R_2 = \sum_{i=0}^{n-1} r_{2i}2^i = r_{2(n-1)} \cdots r_{21}r_{20}$;
   c) $R_3 = \sum_{i=0}^{n-1} r_{3i}2^i = r_{3(n-1)} \cdots r_{31}r_{30}$.
6) $\hat{m}_i = (M/m_i)$, whereas, $\langle (1/\hat{m}_i) \rangle_{m_i}$ is the multiplicative inverse of $\hat{m}_i$ (i.e., $\langle \hat{m}_i \langle (1/\hat{m}_i) \rangle_{m_i} \rangle_{m_i} = 1$).
7) $\lfloor . \rfloor$ is the floor value of (.) (i.e., the largest integer equal to or less than (.).
8) Logic operators: NOT, AND, OR, and XOR are given by the symbols $\overline{(.)}$, $\wedge$, $\vee$, and $\oplus$, respectively.
9) Modulo (2^v-1) multiplication of w by 2^q is obtained by circulating the binary value of w to the left q bits. However, modulo (2^v-1) of a negative number $-w$ is obtained taking the one's complement of w.

Section II will introduce the conversion algorithm and the hardware structure of the proposed converter. However, Section III will evaluate and compare the performance of the proposed converter with other similar ones.

Manuscript received July 28, 2016; accepted September 8, 2016. Date of publication September 12, 2016; date of current version July 31, 2017. This brief was recommended by Associate Editor M. Alioto.

The author is with the Computer Engineering Department, Princess Sumaya University for Technology, Amman 11941, Jordan (e-mail: a.hiasat@psut.edu.jo).

Digital Object Identifier 10.1109/TCSII.2016.2608335

II. Residue-to-Binary Converter Structure for the Moduli Set $(2^{n+1}-1, 2^n, 2^n-1)$

A. Conversion Algorithm

For the three-moduli set under consideration, $\hat{m}_1 = m_2 m_3 = 2^n(2^n-1)$, $\hat{m}_2 = m_1 m_3 = (2^{n+1}-1)(2^n-1)$, and $\hat{m}_3 = m_1 m_2 = 2^n(2^{n+1}-1)$.

The multiplicative inverses of this moduli set are [10]: $\langle(1/\hat{m}_1)\rangle_{m_1} = \langle -4\rangle_{m_1}$, $\langle(1/\hat{m}_2)\rangle_{m_2} = 1$, and $\langle(1/\hat{m}_3)\rangle_{m_3} = 1$. The conversion algorithm adopted in this brief is similar to the approach adopted in [8]. The value X can be recovered from its residue representation using the general formula [1]

$$X = \left\lfloor \frac{X}{a} \right\rfloor a + \langle X \rangle_a \tag{1}$$

where a is a positive integer. Using $a = m_2 m_3 = \hat{m}_1$, then

$$X = \left\lfloor \frac{X}{\hat{m}_1} \right\rfloor m_2 m_3 + \langle X \rangle_{\hat{m}_1}. \tag{2}$$

Dividing (2) by 2^n and taking the floor value

$$\left\lfloor \frac{X}{2^n} \right\rfloor = \left\lfloor \frac{X}{\hat{m}_1} \right\rfloor \left(2^n - 1\right) + \left\lfloor \frac{\langle X \rangle_{\hat{m}_1}}{2^n} \right\rfloor. \tag{3}$$

Rearranging terms in (3)

$$\left\lfloor \frac{X}{2^n} \right\rfloor = \left(\overbrace{\left\lfloor \frac{X}{\hat{m}_1} \right\rfloor}^{n+1} 2^n \star \overbrace{\left\lfloor \frac{\langle X \rangle_{\hat{m}_1}}{2^n} \right\rfloor}^{n} \right) - \overbrace{\left\lfloor \frac{X}{\hat{m}_1} \right\rfloor}^{n+1} \tag{4}$$

where $\star$ is the concatenation operator.
Applying (1) once more, for $a = 2^n$

$$X = \left\lfloor \frac{X}{2^n} \right\rfloor 2^n + R_2. \tag{5}$$

Substituting (4) into (5)

$$X = \overbrace{\left(\left\lfloor \frac{X}{\hat{m}_1} \right\rfloor 2^n \star \left\lfloor \frac{\langle X \rangle_{\hat{m}_1}}{2^n} \right\rfloor - \left\lfloor \frac{X}{\hat{m}_1} \right\rfloor \right)}^{2n+1} 2^n + R_2. \tag{6}$$

The last equation suggests that X can be evaluated as follows. First, compute $\lfloor (X/\hat{m}_1) \rfloor$ and $\lfloor (\langle X \rangle_{\hat{m}_1}/2^n) \rfloor$. Then, subtract $\lfloor (X/\hat{m}_1) \rfloor$ from the concatenated value $\lfloor (X/\hat{m}_1) \rfloor \star \lfloor (\langle X \rangle_{\hat{m}_1}/2^n) \rfloor$. X is obtained by concatenating the subtraction result to R_2, where R_2 constitutes the least significant part.

In the following sections, the terms $\lfloor (X/\hat{m}_1) \rfloor$ and $\lfloor (\langle X \rangle_{\hat{m}_1}/2^n) \rfloor$ are evaluated sequentially.

B. Computing $\lfloor (X/\hat{m}_1) \rfloor$

The binary equivalent of a residue value is obtained by applying the Chinese remainder theorem (CRT), defined as [1]

$$X = \sum_{i=1}^{L} \hat{m}_i \left\langle \frac{1}{\hat{m}_i} \right\rangle_{m_i} R_i \; - M\, I(X) \tag{7}$$

where $I(X)$ is a non-negative integer that depends on X.

Substituting $\hat{m}_1$, $\hat{m}_2$, and $\hat{m}_3$ and the corresponding multiplicative inverses into (7), leads to

$$X = \langle -4 \rangle_{m_1} 2^n \left(2^n - 1\right) R_1 + \left(2^{n+1} - 1\right)\left(2^n - 1\right) R_2 + 2^n \left(2^{n+1} - 1\right) R_3 - M\, I(X). \tag{8}$$

Dividing (8) by $\hat{m}_1 = 2^n(2^n - 1)$

$$\frac{X}{\hat{m}_1} = \langle -4 \rangle_{m_1} R_1 + \left(2 - \frac{1}{2^n}\right) R_2 + \left(2 + \frac{1}{2^n - 1}\right) R_3 - m_1 I(X). \tag{9}$$

Taking the floor value of both sides of (9) and applying modulus m_1

$$\left\langle \left\lfloor \frac{X}{\hat{m}_1} \right\rfloor \right\rangle_{m_1} = \left\langle \left\lfloor \langle -4 \rangle_{m_1} R_1 + \left(2 - \frac{1}{2^n}\right) R_2 + \left(2 + \frac{1}{2^n - 1}\right) R_3 - m_1 I(X) \right\rfloor \right\rangle_{m_1}. \tag{10}$$

Since $\lfloor (X/\hat{m}_1) \rfloor < m_1$, then

$$\left\langle \left\lfloor \frac{X}{\hat{m}_1} \right\rfloor \right\rangle_{m_1} = \left\lfloor \frac{X}{\hat{m}_1} \right\rfloor. \tag{11}$$

Hence, (10) can be rewritten as

$$\begin{aligned} \left\lfloor \frac{X}{\hat{m}_1} \right\rfloor &= \left\langle \left\lfloor \langle -4 \rangle_{m_1} R_1 + \left(2 - \frac{1}{2^n}\right) R_2 + \left(2 + \frac{1}{2^n - 1}\right) R_3 - m_1 I(X) \right\rfloor \right\rangle_{m_1} \\ &= \left\langle \left\lfloor \langle -4 \rangle_{m_1} R_1 + 2R_2 + 2R_3 + \frac{R_3}{2^n - 1} - \frac{R_2}{2^n} - m_1 I(X) \right\rfloor \right\rangle_{m_1} \\ &= \left\langle \langle -4 \rangle_{m_1} R_1 + 2R_2 + 2R_3 + \left\lfloor \frac{R_3}{2^n - 1} - \frac{R_2}{2^n} \right\rfloor - m_1 I(X) \right\rangle_{m_1} \\ &= \left\langle \langle -4R_1 \rangle_{m_1} + \langle 2R_2 \rangle_{m_1} + \langle 2R_3 \rangle_{m_1} + \left\lfloor \frac{R_3}{2^n - 1} - \frac{R_2}{2^n} \right\rfloor \right\rangle_{m_1}. \end{aligned} \tag{12}$$

Defining $U = \langle \lfloor (R_3/2^n - 1) - (R_2/2^n) \rfloor \rangle_{m_1}$, then it can be easily verified that $U = 0$, if $R_3 \geq R_2$. Otherwise, $U = \langle -1 \rangle_{m_1} = (2^{n+1} - 2)$. Hence, U can be written as

$$U = \begin{cases} 0, & \text{if } R_3 \geq R_2 \\ 2^{n+1} - 2, & \text{if } R_3 < R_2. \end{cases} \tag{13}$$

Therefore, (12) can be written as

$$\left\lfloor \frac{X}{\hat{m}_1} \right\rfloor = \left\langle \widetilde{R}_1 + \widetilde{R}_2 + \widetilde{R}_3 + U \right\rangle_{m_1} \tag{14}$$

where, $\widetilde{R}_1 = \langle -4R_1 \rangle_{m_1}$, $\widetilde{R}_2 = \langle 2R_2 \rangle_{m_1}$, and $\widetilde{R}_3 = \langle 2R_3 \rangle_{m_1}$.

Using $(2^v - 1)$ properties and since $R_1 = \overbrace{r_{1n}r_{1(n-1)}\cdots r_{11}r_{10}}^{n+1}$, then

$$\widetilde{R}_1 = \overbrace{\bar{r}_{1(n-2)}\bar{r}_{1(n-3)}\cdots\bar{r}_{10}\bar{r}_{1n}\bar{r}_{1(n-1)}}^{n+1}. \quad (15)$$

Since $R_2 = \overbrace{r_{2(n-1)}\cdots r_{21}r_{20}}^{n}$, then $\langle 2R_2\rangle_{m_1} = \overbrace{r_{2(n-1)}\cdots r_{21}r_{20}\,0}^{n+1}$. Hence, the binary representation of $\widetilde{R}_2$ is

$$\widetilde{R}_2 = \overbrace{r_{2(n-1)}\cdots r_{21}r_{20}\,0}^{n+1}. \quad (16)$$

Similarly, since $R_3 = \overbrace{r_{3(n-1)}\cdots r_{31}r_{30}}^{n}$, while $\langle 2R_3\rangle_{m_1} = \overbrace{r_{3(n-1)}\cdots r_{31}r_{30}\,0}^{n+1}$. Thus

$$\widetilde{R}_3 = \overbrace{r_{3(n-1)}\cdots r_{31}r_{30}\,0}^{n+1}. \quad (17)$$

C. Computing $\lfloor(\langle X\rangle_{\hat{m}_1}/2^n)\rfloor$

The term $\lfloor(\langle X\rangle_{\hat{m}_1}/2^n)\rfloor$ is computed by evaluating, first, $\langle X\rangle_{\hat{m}_1}$, then taking the floor value after dividing it by 2^n.

Defining a two-moduli subset $(2^n, 2^n - 1)$, where $m_2 = 2^n$ and $m_3 = 2^n - 1$, then $\hat{m}_2 = (2^n - 1)$ and $\hat{m}_3 = 2^n$. It can also be easily proved that $\langle(1/\hat{m}_2)\rangle_{m_2} = \langle -1\rangle_{m_2}$ and $\langle(1/\hat{m}_3)\rangle_{m_3} = 1$. Applying the CRT to this submoduli set

$$\langle X\rangle_{m_2m_3} = \left\langle -\left(2^n - 1\right)R_2 + 2^nR_3\right\rangle_{m_2m_3}. \quad (18)$$

Simplifying (18)

$$\langle X\rangle_{m_2m_3} = \left\langle\, 2^n(R_3 - R_2) + R_2\right\rangle_{m_2m_3}. \quad (19)$$

A basic identity in RNS is given by: $\langle aP_1\rangle_{P_1P_2} = P_1\langle a\rangle_{P_2}$, where a, P_1, and P_2 are positive integers [1]. Applying this identity to (19) leads to: $\langle 2^n(R_3 - R_2)\rangle_{m_2m_3} = \langle m_2 \quad (R_3 - R_2)\rangle_{m_2m_3} = m_2\langle R_3 - R_2\rangle_{m_3}$. This implies that: $(2^n\langle R_3 - R_2\rangle_{m_3} + R_2) < m_2m_3$.

Therefore, (19) can be rewritten as

$$\langle X\rangle_{m_2m_3} = 2^n\langle R_3 - R_2\rangle_{m_3} + R_2. \quad (20)$$

Dividing (20) by 2^n, taking the floor value of both sides and recalling that $m_2m_3 = \hat{m}_1$

$$\left\lfloor \frac{\langle X\rangle_{\hat{m}_1}}{2^n} \right\rfloor = \langle R_3 - R_2\rangle_{m_3}. \quad (21)$$

D. Numerical Example

Given the moduli set (63, 32, 31) where $n = 5$, it is required to convert the RNS value of $X = (40, 25, 21)$ to its binary value.

In the following, the subscript 2 denotes a binary number.

$R_1 = (101000)_2$, using (15), $\widetilde{R}_1 = (011\ 101)_2 \xrightarrow{\text{decimal}} = 29$.

$R_2 = (11001)_2$, using (16), $\widetilde{R}_2 = (110\ 010)_2 \xrightarrow{\text{decimal}} = 50$.

$R_3 = (10101)_2$, using (17), $\widetilde{R}_3 = (101\ 010)_2 \xrightarrow{\text{decimal}} = 42$.

Since $R_3 < R_2$ and using (13), $U = 2^{n+1} - 2 = 62$.

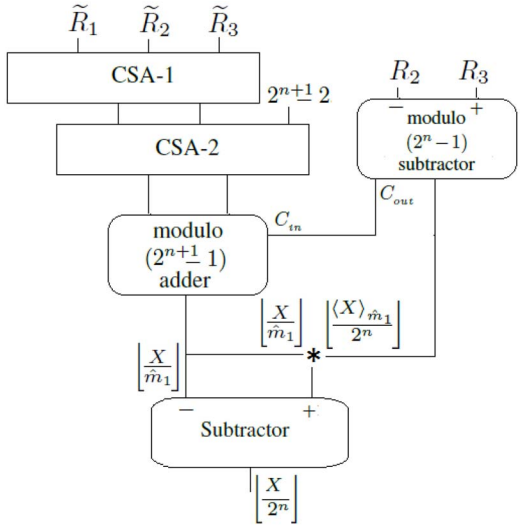

Fig. 1. Block diagram of the proposed converter for the moduli set $(2^{n+1}-1, 2^n, 2^n - 1)$.

Using (14), $\lfloor(X/\hat{m}_1)\rfloor = \langle\widetilde{R}_1 + \widetilde{R}_2 + \widetilde{R}_3 + U\rangle_{m_1} = \langle 29 + 50 + 42 + 62\rangle_{63} = 57$.

Using (21), $\lfloor(\langle X\rangle_{\hat{m}_1}/2^n)\rfloor = \langle R_3 - R_2\rangle_{m_3} = \langle 21 - 25\rangle_{31} = 27$.

Using (4), $\lfloor X/2^n\rfloor = (57 \times 32 + 27) - 57 = 1794$.

Using (5), $X = (1794 \times 32) + 25 = 57433$.

Repeating but for $X = (40, 21, 25)$. $\widetilde{R}_1 = 29$, $\widetilde{R}_2 = 42$, $\widetilde{R}_3 = 50$, and $U = 0$. Therefore, $\lfloor(X/\hat{m}_1)\rfloor = \langle 29 + 42 + 50 + 0\rangle_{63} = 58$. $\lfloor(\langle X\rangle_{\hat{m}_1}/2^n)\rfloor = \langle 25 - 21\rangle_{31} = 4$. $\lfloor X/2^5\rfloor = (58 \times 32 + 4) - 58 = 1802$. $X = (1802 \times 32) + 21 = 57685$.

III. Hardware Implementation

Fig. 1 shows the block diagram of the proposed reverse converter. It consists of two carry-save-adders (CSA). CSA-1, which consists of n full-adders (FA), adds the three $(n+1)$-bit variables, $\widetilde{R}_1$, $\widetilde{R}_2$, and $\widetilde{R}_3$ (the least significant bits in $\widetilde{R}_2$ and $\widetilde{R}_3$ are zeros). However, CSA-2 adds the output of CSA-1 to $U = 2^{n+1} - 2$, where it is assumed that $R_3 < R_2$. If it turns out that this is not the case, a $(+1)$ will be added later on. Since $(2^{n+1} - 2)$ is a constant, CSA-2 consists of one half-adder (HA) and n HA-like (HAL) circuits. An HAL is a circuit that adds two bits, while assuming that the third bit is 1. The HAL consists of an XNOR gate to produce the sum and an OR gate to produce the carry. The output of CSA-2 is applied to a modulo $(2^{n+1} - 1)$ adder to compute $\lfloor(X/\hat{m}_1)\rfloor$. The modulo $(2^n - 1)$ subtractor in Fig. 1 realizes (21). This subtractor receives R_2 and R_3, where each consists of n bits, to compute $\lfloor(\langle X\rangle_{\hat{m}_1}/2^n)\rfloor = \langle R_3 - R_2\rangle_{(2^n-1)}$. As shown in Fig. 1, the modulo $(2^n - 1)$ subtractor operates in parallel with the circuit needed to compute $\lfloor(X/\hat{m}_1)\rfloor$, thus, it adds no delay. The subtractor also produces an output carry C_{out}. The value of C_{out} is 1 if $(R_3 - R_2) \geq 0$. Otherwise, $C_{\text{out}} = 0$. This C_{out} is fed as an input carry C_{in} to modulo $(2^{n+1} - 1)$ adder. This, in fact, rectifies the previous assumption about $U = -1$.

The binary modulo 2^{2n+1} subtractor subtracts $\lfloor(X/\hat{m}_1)\rfloor$ from the concatenated value $\lfloor(X/\hat{m}_1)\rfloor \star \lfloor(\langle X\rangle_{\hat{m}_1}/2^n)\rfloor$ to produce $\lfloor X/2^n\rfloor$. It is worth pointing out that just the least significant $(n + 1)$ bits of the subtractor are fully utilized. Therefore, to realize the modulo 2^{2n+1} subtractor, a modulo

2^{n+1} subtractor is used along with a n bit circuit that propagates the carry resulting from the modulo 2^{n+1} subtractor.

IV. Evaluation, Comparison, and VLSI Realization

A. Theoretical Evaluation

The unit-gate model presented in [17]–[18] is used to evaluate the area and time requirements of the proposed converter and other competitive ones. The unit-gate model considers a two-input monotonic gate to have the area of one unit and the delay of one unit. The model also considers the XOR and XNOR gates to have the area of two units and the delay of two units, but ignores all inverters. The FA has the area of seven units and the delay of four units. However, an HA and an HAL have the area of three units and delay of two units. Furthermore, it is assumed that a (3×1) multiplexer (MUX) consists of two consecutive (2×1) MUXs, where a (2×1) MUX has the same complexity as an HA/HAL.

Moreover, modulo (2^k-1) and modulo 2^k adders, presented and referred to in [18]–[21], will be used in all the designs evaluated in this section. The area and time delay of a modulo (2^k-1) adder are, respectively [18]–[21]: $(3k\lceil\log_2 k\rceil+5k)$ and $(2\lceil\log_2 k\rceil+3)$. However, the area and time delay of a modulo 2^k adder are, respectively [18]–[21]: $((3/2)\lceil\log_2\ k\rceil+5k)$ and $(2\ \lceil\log_2\ k\rceil+3)$.

Additionally, in all structures under consideration, it will be assumed that n is not a power of 2. To further simplify the analysis, all $(n+1)$ bit adders will be assumed to have the complexity of n bit adders.

1) Area and Time Requirements of the Proposed Converter: As shown in Fig. 1, the proposed converter consists of a modulo $(2^{n+1}-1)$ adder, a modulo (2^n-1) subtractor, a modulo 2^{2n+1} subtractor and two CSAs. CSA1 has the area and delay of $7n$ and 4, respectively. While, CSA2 has the area and delay of $3(n+1)$ and 2, respectively. The modulo $(2^{n+1}-1)$ adder has the area and delay of $(3(n+1)\lceil\log_2 n\rceil+5(n+1))$, and $2\lceil\log_2 n\rceil+3$, respectively. Similarly, the modulo (2^n-1) subtractor has the area and delay of $(3n\lceil\log_2 n\rceil+5n)$, and $(2\lceil\log_2 n\rceil+3)$, respectively. However, because it is subtracting a $(n+1)$-bit variable from a $(2n+1)$ bit variable, the modulo 2^{2n+1} subtractor in Fig. 1 consists of two sub-circuits: a modulo 2^{n+1} subtractor and a carry-propagate circuit. The 2^{n+1} subtractor has the area and delay of $(3/2)(n+1)\lceil\log_2 n\rceil+5(n+1)$, and $2\lceil\log_2 n\rceil+3$, respectively. While the carry-propagate circuit has an area of $5n$ and has no delay [19]–[21]. Enforcing the above simplification assumption, the area and delay of the proposed converter are, $(7.5n\lceil\log_2 n\rceil+30n)$ and $(4\lceil\log_2 n\rceil+12)$, respectively.

2) Area and Time Requirements of the Converters in [10]: The converter in [10] deals with the same moduli set considered in this brief. Three converters were presented for the same moduli set [10]. The first converter, (denoted as converter 1), was based on mixed-radix conversion (MRC), while the other two were based on CRT (denoted as converters 2 and 3). converter 2 has almost the same complexity as converter 3. Therefore, just converters 1 and 2 will be considered for comparison.

TABLE I
Unit-Gate Model Results for Reverse Converters of the Moduli Sets: $(2^{n+1}-1, 2^n, 2^n-1)^\star$ and $(2^{n-1}-1, 2^n, 2^n-1)^{\star\star}$

Converter	Area	Delay
Proposed*	$7.5n\lceil log_2 n\rceil+30n$	$4\lceil log_2 n\rceil+12$
Converter 1* [10]	$10.5n\lceil log_2 n\rceil+25n$	$6\lceil log_2 n\rceil+9$
Converter 2* [10]	$9n\lceil log_2 n\rceil+113n$	$2\lceil log_2 n\rceil+25$
Converter** [9]	$10.5n\lceil log_2 n\rceil+44n$	$4\lceil log_2 n\rceil+20$

Converter 1 consists of two modulo $(2^{n+1}-1)$ subtractors, one modulo (2^n-1) and one modulo 2^{2n+1} subtractor. This last subtractor is similar to the last subtractor in the proposed converter. Therefore, the area and delay of converter 1, under the same assumption, are, $(10.5n\lceil\log_2 n\rceil+25n)$ and $(6\lceil\log_2 n\rceil+9)$, respectively.

Converter 2 consists of five $(2n+3)$ CSAs (where, mostly, FA are used), three modulo 2^{2n+3} adders and $(2n+1)$ blocks of (3×1) MUXs. To further simplify the evaluation, it is assumed that all $(2n+3)$-bit adders have the same complexity as $2n$-bit adders. Hence, the area and delay of converter 2 are: $(9n\lceil\log_2 n\rceil+113n)$ and $(2\lceil\log_2 n\rceil+25)$, respectively.

3) Area and Time Requirements of the Converter in [9]: The converter presented in [9] is concerned with the moduli set $(2^{n-1}-1, 2^n, 2^n-1)$. This set provides a dynamic range of $(3n-1)$. However, the three moduli have similar forms like the ones in the moduli set $(2^{n+1}-1, 2^n, 2^n-1)$. Therefore, it can be reasonably compared with the proposed converter. The converter in [9] requires $(n-1)$ blocks of (2×1) MUXs, four $(n-1)$-bit CSAs, two modulo $(2^{n-1}-1)$ adders, one modulo (2^n-1) subtractor and one modulo 2^{2n+1} similar to the last subtractor in the proposed converter. The area and delay of the converter in [9] are, respectively, are: $(10.5\lceil\log_2 n\rceil+44n)$ and $(4\lceil\log_2 n\rceil+20)$.

The results of applying the unit-gate model to the proposed converter and the other three ones are summarized in Table I.

B. VLSI Synthesis Evaluation

The four converters under comparison; the proposed one, converter 1 [10], converter 2 [10] and the converter presented in [9], have all been modeled in Verilog HDL using Synopsys Design compiler (G-2012.06). These converters have been mapped to 65 nm Synopsys designware logic libraries. Synopsys IC compiler has been used to perform the place and route layout phase. The power consumed was calculated using Synopsys power compiler. Using Synopsys simulator, the functionality and correctness of the four designs have been checked. In all the designs under consideration in this section, the fast parallel-prefix modulo 2^k and modulo (2^k-1) adders presented in [18]–[21] have been realized. Table II summarizes the synthesis results of the four converters after place and route phase. However, Table III shows the percentages of improvements, which the proposed converter has achieved for values of $n = 7, 11, 15, 18,$ and 22. The reductions

TABLE II
SYNTHESIS RESULTS FOR REVERSE CONVERTERS OF MODULI SETS: $(2^{n+1}-1, 2^n, 2^n-1)^\star$ AND $(2^{n-1}-1, 2^n, 2^n-1)^{\star\star}$

Converter	n	Area μm^2	Delay ps	Power μW
Proposed*	7	1590.5	851.2	61.0
	11	2881.2	991.8	78.4
	15	3958.4	1026.9	89.3
	18	5294.6	1153.4	103.5
	22	7017.2	1172.7	145.9
Converter 1*, [10]	7	1802.9	1015.2	69.6
	11	3297.2	1172.3	87.2
	15	4479.2	1191.5	96.8
	18	5985.5	1354.6	112.4
	22	7845.3	1392.1	162.2
Converter 2*, [10]	7	3867.5	1092.0	105.3
	11	7221.6	1242.2	155.4
	15	9684.2	1264.5	185.1
	18	12289.0	1392.0	215.9
	22	17035.7	1422.4	294.3
Converter** [9]	7	2256.5	1095.8	70.4
	11	4067.9	1326.4	93.5
	15	5451.1	1361.5	107.1
	18	7053.5	1504.0	129.8
	22	9095.6	1530.2	173.0

TABLE III
AREA AND DELAY REDUCTIONS OF THE PROPOSED CONVERTER COMPARED WITH OTHER COMPETITIVE ONES

Proposed Converter compared with:	n	(%) Area Reduction	(%) Delay Reduction	(%) Power Reduction
Converter 1, [10]	7	11.8	16.2	12.4
	11	12.6	15.4	10.1
	15	11.6	13.8	7.7
	18	11.5	14.9	7.9
	22	10.6	15.8	10.0
Converter 2, [10]	7	58.9	22.1	42.1
	11	60.1	20.2	49.6
	15	59.1	18.8	51.8
	18	56.9	17.1	52.1
	22	58.8	17.6	50.4
Converter [9]	7	29.5	22.3	13.4
	11	29.2	25.2	16.1
	15	27.4	24.6	16.6
	18	24.9	23.3	20.3
	22	22.9	23.4	15.7

in area, delay and power are computed using the formula: $[(\text{competitive} - \text{proposed})/\text{competitive}] \times 100\%$.

Compared with converter 1 [10], which is an MRC-based, the new converter has achieved area, delay, and power reductions of $(10.6 - 12.6)\%$, $(13.8 - 16.2)\%$, and $(7.7 - 12.4)\%$, respectively. When compared with Converter 2 [10], which is a CRT-based, the proposed structure, which is also a CRT-based, has achieved area, delay, and power reductions of $(56.9 - 60.1)\%$, $(17.1 - 22.1)\%$, and $(42.1 - 52.1)\%$, respectively. As compared with the converter presented in [9], the new designed converter has reduced area by $(22.9 - 29.5)\%$, time by $(22.3 - 25.2)\%$, and power by $(13.4 - 20.3)\%$.

V. CONCLUSION

This brief presented a residue-to-binary converter for the moduli set $(2^{n+1} - 1, 2^n, 2^n - 1)$. When compared with published work for similar sets of the form $(2^{n\pm1} - 1, 2^n, 2^n - 1)$, the proposed converter has demonstrated area improvements of $(10.6 - 60.1)\%$, time improvements of $(13.8 - 25.2)\%$, and power improvements by $(7.7 - 52.1)\%$.

REFERENCES

[1] N. S. Szabó and R. I. Tanaka, *Residue Arithmetic and Its Applications to Computer Technology*. New York, NY, USA: McGraw-Hill, 1967.

[2] M. A. Soderstrand, G. A. Jullien, F. J. Taylor, and K. W. Jenkins, Eds., *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*. New York, NY, USA: IEEE Press, 1986.

[3] R. Chaves and L. Sousa, "RDSP: A RISC DSP based on residue number system," in *Proc. Symp. Digit. Syst. Design Architect. Methods Tools*, Sep. 2003, pp. 128–135.

[4] T. Toivonen and J. Heikkil, "Video filtering with Fermat number theoretic transforms using residue number system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 1, pp. 92–101, Jan. 2006.

[5] J.-C. Bajard and L. Imbert, "A full RNS implementation of RSA," *IEEE Trans. Comput.*, vol. 53, no. 6, pp. 769–774, Jun. 2004.

[6] P. Bernardson, "Fast memoryless, over 64 bits, residue-to-binary convertor," *IEEE Trans. Circuits Syst.*, vol. 32, no. 3, pp. 298–300, Mar. 1985.

[7] Y. Wang, X. Song, M. Aboulhamid, and H. Shen, "Adder based residue to binary number converters for $(2^n-1, 2^n, 2^n+1)$," *IEEE Trans. Signal Process.*, vol. 50, no. 7, pp. 1772–1779, Jul. 2002.

[8] A. A. Hiasat and S. H. Abdel-Aty-Zohdy, "Residue-to-binary arithmetic converter for the moduli set $(2^k, 2^k-1, 2^{k-1}-1)$," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 45, no. 2, pp. 204–209, Feb. 1998.

[9] W. Wang, M. N. S. Swamy, M. O. Ahmad, and Y. Wang, "A high-speed residue-to-binary converter for the three-moduli $(2^k, 2^k-1, 2^{k-1}-1)$ RNS and a scheme for its VLSI implementation," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 12, pp. 1576–1581, Dec. 2000.

[10] P. V. A. Mohan, "RNS-to-binary converter for a new three-moduli set $(2^{n+1}-1, 2^n, 2^n-1)$," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 9, pp. 775–779, Sep. 2007.

[11] L. Sousa and S. Anto, "MRC-based RNS reverse converters for the four-moduli sets $(2^n+1, 2^n-1, 2^n, 2^{2n+1}-1)$ and $(2^n+1, 2^n-1, 2^{2n}, 2^{2n}-1)$," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 4, pp. 244–248, Apr. 2012.

[12] H. Pettenghi, R. Chaves, and L. Sousa, "RNS reverse converters for moduli sets with dynamic ranges up to $(8n+1)$-bit," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 6, pp. 1487–1500, Jun. 2013.

[13] P. Patronik and S. J. Piestrak, "Design of reverse converters for general RNS moduli set $(2^k, 2^n-1, 2^n+1, 2^{n+1}-1)$ and $(2^k, 2^n-1, 2^n+1, 2^n+1)(n \text{ even})$," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 6, pp. 1687–1700, Jun. 2014.

[14] P. Patronik and S. J. Piestrak, "Design of reverse converters for the new RNS moduli set $(2^n+1, 2^n-1, 2^n, 2^{n-1}+1)$ and $(2^k, 2^n-1, 2^n+1, 2^{n-}+1)(n \text{ odd})$," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 12, pp. 3436–3449, Dec. 2014.

[15] M. Xu, Z. Bian, and R. Yao, "Fast sign detection algorithm for the RNS moduli set $(2^{n+1}-1, 2^n-1, 2^n)$," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 2, pp. 379–383, Feb. 2015.

[16] S. Kumar and C.-H. Chang, "A new fast and area-efficient adder-based sign detector for RNS $2^{n-1}, 2^n, 2^n+1$," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 7, pp. 2608–2612, Jul. 2016, doi: 10.1109/TVLSI.2016.2516522.

[17] R. Zimmermann, "Efficient VLSI implementation of modulo $(2^n \pm 1)$ addition and multiplication," in *Proc. 14th IEEE Symp. Comput. Arithmetic*, Adelaide, SA, Australia, Apr. 1999, pp. 158–167.

[18] A. Tyagi, "A reduced-area scheme for carry-select adders," *IEEE Trans. Comput.*, vol. 42, no. 10, pp. 1163–1170, Oct. 1993.

[19] L. Kalamboukas, D. Nikolos, C. Efstathiou, H. T. Vergos, and J. Kalamatianos, "High-speed parallel-prefix module $2^n - 1$ adders," *IEEE Trans. Comput.*, vol. 49, no. 7, pp. 673–680, Jul. 2000.

[20] H. T. Vergos, C. Efstathiou, and D. Nikolos, "Diminished-one modulo $2^n + 1$ adder design," *IEEE Trans. Comput.*, vol. 51, no. 12, pp. 1389–1399, Dec. 2002.

[21] G. Dimitrakopoulos and D. Nikolos, "High-speed parallel-prefix VLSI ling adders," *IEEE Trans. Comput.*, vol. 54, no. 2, pp. 225–231, Feb. 2005.