Sādhanā (2020) 45:239
https://doi.org/10.1007/s12046-020-01474-3

An improved RNS-to-binary converter for 7-modulus set $\{2^{n-5}-1, 2^{n-3}-1, 2^{n-2}+1, 2^{n-1}-1, 2^{n-1}+1, 2^n, 2^n+1\}$ for n even

M V N MADHAVI LATHA[1], RASHMI RAMESH RACHH[2] and P V ANANDA MOHAN[3,*]

[1]Department of Electronics and Communication Engineering, Gitam School of Technology, Hyderabad 502329, India
[2]Department of Computer Science and Engineering, Centre for PG Studies, Belagavi 590018, India
[3]Sahakaranagara, Bangalore 560092, India
e-mail: madhavisravan@gmail.com; rashmirachh@gmail.com; anandmohanpv@live.in

MS received 24 April 2020; revised 23 May 2020; accepted 5 July 2020

Abstract. In this paper a residue-to-binary converter for the 7-modulus set $\{2^{n-5}-1, 2^{n-3}-1, 2^{n-2}+1, 2^{n-1}-1, 2^{n-1}+1, 2^n, 2^n+1\}$ for n even is presented, which is an improved version of a reverse converter presented recently. The proposed converter needs less hardware resources ranging from 52.9% to 66.9%, conversion time of 68–70.7% and power dissipation of 47–63% of those needed for an earlier described converter for the same considered dynamic ranges.

Keywords. Residue number systems; reverse converters; three-modulus sets; CRT; mixed radix conversion.

1. Introduction

Residue number system (RNS) with high dynamic range needed in applications like Digital Signal processing and cryptography can be realized using a set of mutually prime moduli [1, 2]. Several 3-modulus sets have been studied in literature ($\{2^n-1, 2^{n+k}, 2^n+1\}$ [1, 3] for $0 \le k \le n$, $\{2m-1, 2m, 2m+1\}$ [1, 4] and $\{2^{n+k}, 2^n-1, 2^{n+1}-1\}$ [5] for $0 \le k \le n$) for realizing a dynamic range DR of $3n$–$4n$ bits. Higher DR up to $5n$ bits can be realized by choosing four or more mutually prime moduli of about similar word length from the various options available: 2^n-1, 2^{n+k}, 2^n+1, $2^{n\pm1}\pm1$, $2^{2n}+1$, $2^n \pm 2^{(n+1)/2}+1$ (for n odd), etc. In a recent paper [6] a reverse converter for a new 7-modulus set $\{2^{n-5}-1, 2^{n-3}-1, 2^{n-2}+1, 2^{n-1}-1, 2^{n-1}+1, 2^n, 2^n+1\}$ with $n = 2k$, $k = 4, 5, 6\ldots$that has a DR of $(7n-12)$ bits obtained by deleting modulus $(2^{n-3}+1)$, which has a common factor with $(2^{n-1}+1)$ in the 8-modulus set proposed by Skavantzos *et al* in [7], has been described. In the present paper we describe an alternative approach for the design of reverse converter shown in figure 1, in which two improvements are made over the reverse converter in [6]: (a) the design of 4-modulus converter uses a [3,1] architecture instead of [2,2] architecture in order to reduce hardware resource requirement and conversion time and (b) the complex computations needed in the converter in [6] due to the large value of the multiplicative inverse needed are replaced with operations on smaller numbers. As an illustration for $n = 8$, 10, 12 and 14, the multiplicative inverses are 12406, 583440, 12846866 and 843514373, respectively, resulting in complex modulo M_{123} multipliers where $M_{123} = (2^{n-5}-1)(2^{n-3}-1)(2^{n-2}+1)$.

The rest of the paper is organized as follows. In section 2 the proposed RNS-to-binary converter for the 7-modulus set is described and in section 3 the evaluation of the hardware resource requirement and conversion time and comparison with the converter in [6] are presented together with ASIC implementation results. Section 4 concludes the paper.

2. Proposed reverse converter

In the proposed architecture of figure 1, we compute the decoded number X'' corresponding to the residues x_4, x_5, x_6 and x_7 pertaining, respectively, to moduli $m_4 = 2^{n-1}-1$, $m_5 = 2^{n-1}+1$, $m_6 = 2^n$, $m_7 = 2^n+1$ using a 2-stage 4-modulus converter in which the first stage considers moduli $\{x_4, x_5, x_7\}$ and computes the MRC digits U_C, U_B and U_A using MRC as shown in figure 2a. Note that $U_A = x_7$. The various multiplicative inverses needed in figure 2a can be found easily as

$$\begin{aligned} X_B &= \left(\frac{1}{m_7}\right) \bmod m_5 = -1, \\ X'_C &= \left(\frac{1}{m_7}\right) \bmod m_4 = 2^{n-2} + 2^{n-4} + \cdots + 4 + 1, \\ X_C &= \left(\frac{1}{m_5}\right) \bmod m_4 = 2^{n-2}. \end{aligned} \quad (1)$$

*For correspondence
Published online: 17 September 2020

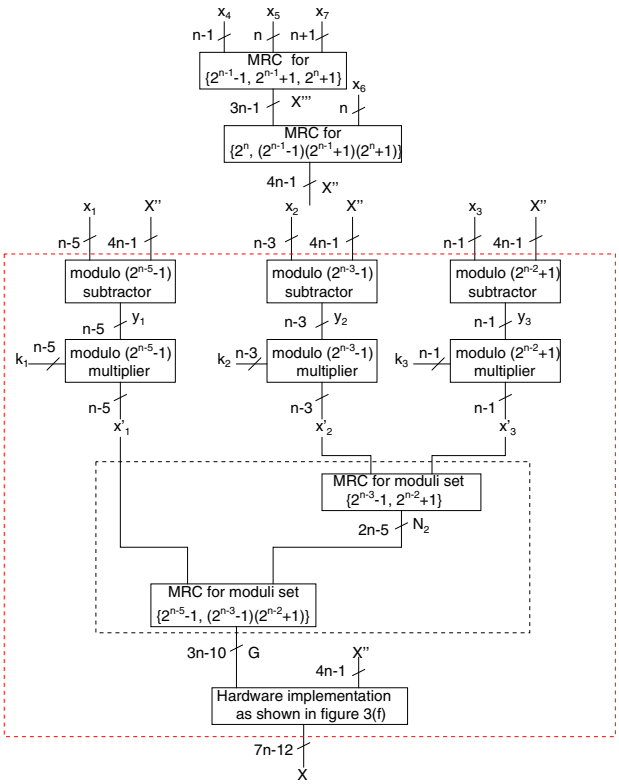

Figure 1. Proposed architecture.

$$
\begin{array}{l}
\underline{2^{n-1}\text{-}1 \qquad 2^{n-1}+1 \quad 2^n+1} \\
x_4x_5x_7(=U_A) \\
\underline{\text{-}x_7\text{-}x_7} \\
(x_4\text{-}x_7)_{m4}(x_5\text{-}x_7)_{m5} \\
\underline{\times X'_C \times X_B} \\
U'_C U_B \\
\underline{\text{-}U_B} \\
(U'_C\text{-}\ U_B)_{m4} \\
\underline{\times X_C} \\
U_C
\end{array}
$$

(a)

$$
\begin{array}{l}
\underline{2^n \quad M_{457}} \\
x_6X''' \\
\underline{\text{-}\,X'''} \\
(x_6\text{-}X''')_{m6} = Y_D \\
\left(\dfrac{1}{M_{457}}\right)_{m_6} = X_D \\
\hline
U_D
\end{array}
$$

(b)

Figure 2. (**a**) MRC procedure for computing X''' and (**b**) MRC procedure for computing X''.

In the implementation of figure 2a, modulo subtractions of words of different bit lengths are needed. They can be performed using the periodic properties of moduli of type (2^x-1) and (2^x+1) [1, 2]. The detailed architectures for computing U_B, U'_C and U_C using these properties are presented in figure 3a–c. The mod$(2^{n-1}-1)$ multiplication with multiplicative inverse X'_C (see (1)) produces $n/2$ left circular shifted (LCS) versions of the word V, which can be added using $(n/2-2)$ stages of $(n-1)$-bit CSA5 followed by CPA4 all with end-around-carry (EAC) to obtain U'_C as shown in figure 3b. Next U_C can be computed as $(U'_C - U_B)$mod m_4, and multiplication mod m_4 with X_C is obtained by left circular shift (bit mapping needing no hardware) as shown in figure 3c. The decoded word corresponding to residues (x_4, x_5, x_7) is given as

$$X''' = U_A + (2^n+1)U_B + (2^n+1)\left(2^{n-1}+1\right)U_C \qquad (2)$$

In order to reduce the conversion time, X''' is not computed. In another MRC stage, from the residues (X''', x_6) corresponding to the moduli set $(m_4m_5m_7, m_6)$, the decoded number X'' is computed as shown in figure 2b. Note that the MRC digit U_D is given as $U_D = (X''' - x_6) \bmod 2^n$ since the multiplicative inverse needed is $X_D = \left(\frac{1}{M_{457}}\right)_{m_6} = -1$. The final decoded word X'' corresponding to the moduli set $\{m_4, m_5, m_6, m_7\}$ can be obtained as

$$\begin{aligned} X'' &= (2^{2n-2}-1)(2^n+1)U_D + X''' \\ &= (2^{2n-2}-1)(2^n+1)U_D + U_A + (2^n+1)U_B \\ &\quad + (2^n+1)\left(2^{n-1}+1\right)U_C \end{aligned} \qquad (3)$$

which is in carry save form as vectors S', C' and another word $(-D_5)$ where D_1–D_5 are as follows:

$$\begin{aligned} &D_1 = U_D \| U_D \| 0^{n-3} \| U_A, \quad D_2 = 0^{2n-2} \| U_B \| U_B, \\ &D_3 = 0^n \| U_C \| U_C \| 0 \| U_C, \quad D_4 = 0^{2n} \| U_C \| 0^{n-1}, \\ &D_5 = U_D \| U_D. \end{aligned} \qquad (4)$$

The architectures for computing U_D, S'' and C'' are shown in figure 3d and e.

Next, we compute from the input residues $\{x_1, x_2, x_3\}$ and X'' (*i.e.* S'', C'' and $-D_5$) the residues of G, *viz.* $\{x'_1, x'_2, x'_3\}$ corresponding to moduli m_1, m_2 and m_3 as

$$\begin{aligned} (G)_{m_i} = x'_i &= \left(\left(x_i - X''_{m_i}\right)\left(\frac{1}{M_{4567}}\right)_{m_i}\right)_{m_i} \\ &= \left(\left(x_i - S''_{m_i} - C''_{m_i} + D_5\right)\left(\frac{1}{M_{4567}}\right)_{m_i}\right)_{m_i} \\ &= \left(y_i\left(\frac{1}{M_{4567}}\right)_{m_i}\right)_{m_i} \end{aligned} \qquad (5)$$

for i = 1, 2 and 3 where X''_{m_i}, S''_{m_i} and C''_{m_i} are, respectively, the residues of X'', S'' and C'' with respect to moduli m_i. Binary-to-RNS conversion needs to be used to obtain residues of S'', C'' and $-D_5$. We consider S'' and C'', which are of length $(4n-2)$ bits, each as l_{1a}, l_{2a} and l_{3a} and D_5 as

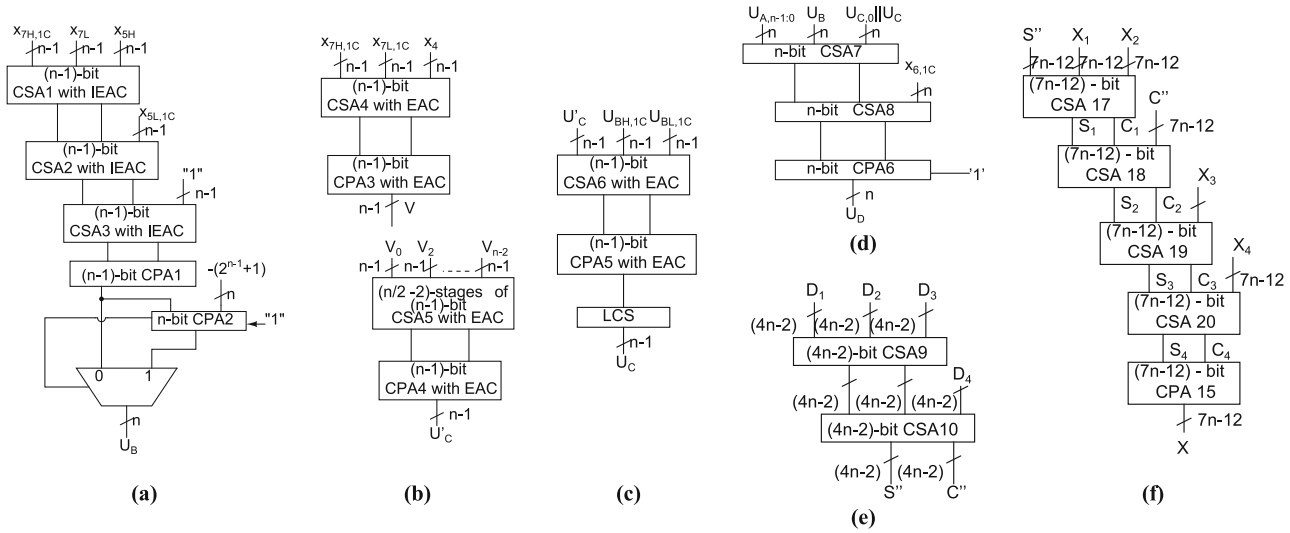

Figure 3. (**a**)–(**f**) Architectures for computation of U_B, U'_C, U_C, U_D, S'', C'' and X.

l_{1b}, l_{2b} and l_{3b} number of (n–5)-bit, (n–3)-bit and (n–2)-bit fields, respectively. One's complement of the $2l_{ia}$ words corresponding to S'' and C'' needs to be added with x_i and l_{ib} words corresponding to D_5, totally using $l_1 = 2l_{1a} + l_{1b} - 1$ level and $l_2 = 2l_{2a} + l_{2b} - 1$ level CSA trees followed by CPA with EAC in case of modulus m_i for $i = 1, 2$ to obtain y_1 and y_2. In the case of modulus m_3, denoting S'' (and C'') as l_{3a} (= 5) number of (n–2)-bit words, *i.e.* $S'' = W_{L_4} W_{L_3} W_{L_2} W_{L_1} W_{L_0}$, we have $(-S'')\text{mod}(2^{n-2}+1) = (-W_{L_4} + W_{L_3} - W_{L_2} + W_{L_1} - W_{L_0}) \bmod (2^{n-2}+1)$. Since $D_5 = D_{51} \| D_{50}$ is of $2n$-bit length, we have $(D_5)\text{mod}(2^{n-2}+1) = (-D_{51}+D_{50})\text{mod}(2^{n-2}+1)$. Thus, the two odd fields and one's complements of the three even fields corresponding to $-S''$, $-C''$ and the two fields D_{50} and $D_{5,1C}$ together with x_3 need to be added using $l_3 = 2l_{3a} + 3$ level CSA tree with inverted EAC and a correction term followed by a $\text{mod}(2^{n-2}+1)$ adder to obtain the residue y_3. The computation of y_1, y_2 and y_3 needs $((l_1+1)(n–5) + (l_2+1)(n–3) + l_3(n–2) + 2n–3)$FA and the computation time is $(2n\text{-}3+l_3)D_{FA}$. Next, three modulo multipliers are needed to compute $(k_i y_i) \bmod m_i$ for $i = 1, 2, 3$ where $k_i = \left(\frac{1}{M_{4567}}\right)_{m_i}$. As an illustration, for $n = 8$, $k_1 = 2$, $k_2 = 6$ and $k_3 = 56$ (corresponding to 12,406 in the technique used in [6]). The modulo multipliers also need to use the periodic properties of moduli. The hardware requirement for computing x'_1, x'_2 and x'_3 from y_1, y_2 and y_3, respectively, are $((n–5)(n–6)+(n–4)(n–3)+n^2–2n+2)$FA+$(2n–6)$HA+XNOR/OR + n(2:1)MUX and the computation time is $(3n–4)D_{FA}$. The 3-modulus two-stage reverse converter described in [6] needs to be used to obtain G from $\{x'_1, x'_2, x'_3\}$ as shown in figure 1. (Note that in this converter in [6], the input residues are x_1, x_2, x_3). The hardware resources needed for this converter are

$$\left(\frac{n^2+n}{2} + n\beta - 3\right) FA + (n-4)HA + (3n-5)XNOR/OR$$

where β is the number of bits that are 1 in the multiplicative inverse needed in the converter $\left(\frac{1}{(2^{n-3}-1)(2^{n-2}+1)}\right)_{(2^{n-5}-1)}$ and the conversion time is $(13.5n–49)D_{FA}$.

Next, the last stage in figure 1 obtains the final ($7n$–12)-bit decoded number as

$$\begin{aligned} X &= X'' + G \times M_{4567} \\ &= X'' + (2^{2n-2} - 1)(2^{2n} + 2^n)G \\ &= S'' + C'' - D_5 + 2^{4n-2}G + 2^{3n-2}G \\ &\quad - 2^{2n}G - 2^nG \end{aligned} \tag{6}$$

using the hardware shown in figure 3f.

3. Evaluation of proposed converter

The hardware requirement for the proposed reverse converter for the 7-modulus set and that described in [6] are presented in Table 1 in terms of unit gates using unit gate model [8] in Table 1. We consider the gates needed for FA, HA, 2:1MUX, EXOR, AND, OR as 7, 3, 3, 2, 1, 1 and delay in terms of unit gate delay Δ_g as 4, 2, 2, 2, 1, 1 units, respectively. It may be noted from Table 1 for n = 8,10 and 12 that the proposed converter needs, respectively, 62%, 55% and 50% of hardware resources and the conversion time needed is about 73% of that needed for the converter in [6].

Table 1. Hardware requirement and conversion time requirements based on unit gate model.

Converter	Total hardware requirement (unit gates)	Conversion time (Δ_g)
[6]	$101.5n^2+n(103.5+21l_2+7l_1)-35l_1-70l_2$ $-973+7\log_2(3n-9)+7\log_2(n+10)^{\dagger}$	$180n-320+4l_2+4\log_2(n+8)+4\log_2(3n-9)$
Proposed converter	$28n^2+n(143+7l_1+7l_2+7l_3+7\beta)$ $-35l_1-21l_2-14l_3-35\beta-163$	$132n-268+4l_3+4\beta$

† l_1 is 2, 4 and 2 and l_2 is 7, 8 and 8, respectively, for n = 8, 10 and 12.

Table 2. Synthesis results of proposed 7-modulus converter.

	Cell area (μm^2)		Delay (ps)		Power (μW)	
n	D12	Proposed	D12	Proposed	D12	Proposed
6	–	–	–	–	–	–
8	5368	3592	10533	7448	1292.499	816.309
10	8472	4893	13682	9732	2704.941	1274.276
12	12255	6491	16962	11548	3744.647	1883.525

The proposed converters for the 7-modulus set for n = 6, 8, 10 and 12 were modelled in Verilog HDL and synthesized using Encounter RTL compiler 14.10 with 45-nm technology library files. Functionality of the converter was verified using NCsim. Synthesis results of the area, power and delay are presented in Table 2, which show that the proposed 7-modulus converter needs less hardware resources ranging from 52.9% to 66.9%, conversion time of 68–70.7% and power dissipation of 47–63% of those needed for the converter in [6] for the considered DRs.

4. Conclusion

In this paper a new residue-to-binary converter for the 7-modulus set $\{2^{n-5}-1, 2^{n-3}-1, 2^{n-2}+1, 2^{n-1}-1, 2^{n-1}+1, 2^n, 2^n+1\}$, which is an improved version of a recently proposed 7-modulus converter, is presented. It is compared with the previously reported converter and the hardware resource requirements and conversion time have been shown to be less than those of the previous design. Vertical extension of the proposed 7-modulus set is being explored to enhance the DR to $8n$–10 and $9n$–10 bits.

References

[1] Soderstrand M A, Jullien G A, Jenkins W K and Taylor F (Eds.) 1986 *Residue number system arithmetic: modern applications in digital signal processing*. IEEE Press, Piscataway, NJ

[2] Ananda Mohan P V 2016 *Residue number systems: theory and applications*. Switzerland: Birkhauser

[3] Chaves R and Sousa L 2004$\{2^n + 1, 2^{n+k}, 2^n - 1\}$: a new RNS moduli set extension. In: *Proceedings of the Euromicro Symposium on Digital System Design*, Rennes, France, pp. 210–217

[4] Sheu M H, Siao S M, Hwang Y T, Sun C C and Lin Y P 2016 New adaptable three-moduli set $\{2^{n+k}, 2^n - 1, 2^{n-1} - 1\}$ for residue number system-based finite impulse response implementation. *IEICE Electronic Express* 13: 1–9

[5] Phalguna P S, Kamath D V and Ananda Mohan P V 2019 Novel RNS-to-binary converters for the three-moduli set $\{2m - 1, 2m, 2m + 1\}$. *Sadhana – Academy Proceedings in Engineering Sciences* 44: 1–10

[6] Madhavilatha M V N, Rachh R R and Ananda Mohan P V 2019 Residue-to-Binary converter for seven moduli set $\{2^{n-5}-1, 2^{n-3}-1, 2^{n-2}+1, 2^{n-1}-1, 2^{n-1}+1, 2^n, 2^n+1\}$. In: *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems*

[7] Skavantzos A, Abdallah M, Stouraitis T and Schinianakis D 2009 Design of a balanced 8-modulus RNS. In: *Proceedings of IEEE ISCAS*, pp. 61–64

[8] Tyagi A 1993 A reduced area scheme for carry-select adders. *IEEE Transactions on Computers* 19: 1163–1170.