Towards Efficient Modular Adders based on Reversible Circuits

Amir Sabbagh Molahosseini
CE Department
Kerman Branch,
Islamic Azad University
Kerman, Iran
amir@iauk.ac.ir

Ailin Asadpoor
CE Department
Kerman Branch,
Islamic Azad University
Kerman, Iran
ailinasadpoor@iauk.ac.ir

Azadeh Alsadat Emrani Zarandi
CE Department
Shahid Bahonar University of Kerman
Kerman, Iran
a.emrani@uk.ac.ir

Leonel Sousa
INESC-ID, Instituto Superior Tecnico (IST),
Universidade de Lisboa
Lisbon, Portugal
leonel.sousa@ist.utl.pt

***Abstract*— Reversible logic is a computing paradigm that has attracted significant attention in recent years due to its properties that lead to ultra-low power and reliable circuits. Reversible circuits are fundamental, for example, for quantum computing. Since addition is a fundamental operation, designing efficient adders is a cornerstone in the research of reversible circuits. Residue Number Systems (RNS) has been as a powerful tool to provide parallel and fault-tolerant implementations of computations where additions and multiplications are dominant. In this paper, for the first time in the literature, we propose the combination of RNS and reversible logic. The parallelism of RNS is leveraged to increase the performance of reversible computational circuits. Being the most fundamental part in any RNS, in this work we propose the implementation of modular adders, namely modulo 2^n-1 adders, using reversible logic. Analysis and comparison with traditional logic show that modulo adders can be designed using reversible gates with minimum overhead in comparison to regular reversible adders.**

Keywords—Residue Number System (RNS); Reversible Circuits; Modular Adder; Parallel-Prefix Adder.

I. INTRODUCTION

Researchers in academia and industry believe that Moore's law is ending, and even newly delivered deep-submicron transistors are not significantly more efficient than their previous generations [1]. Therefore, new computing paradigms should be investigated in order to overcome the predicted performance wall which will be reached in 2020 [1]. This rebooting of computing has to be based on novel methods at different computing levels of design abstraction, including arithmetic and circuit level, in order to address the challenges of the emerging applications such as deep convolutional neural network (DNN) and internet-of-things (IoT) [2]. Residue Number System (RNS) is one unconventional number system [3] that can provide fast and low-power implementation of additions and multiplications. RNS is a different approach of dealing and representing numbers that provide parallelism at arithmetic level [4]. This number system has been applied to achieve parallel and efficient implementations for asymmetric cryptographic and digital signal processing (DSP) [5]. RNS is used nowadays to achieve also energy-efficient and high-performance implementation of various emerging applications, such as deep neural networks, communication networks and cloud storage [3]. However, current implementations of RNS systems, on ASICs and FPGAs, are based on the CMOS technology, which is reaching its limit. Alternative methods and technologies, such as nano-electronic, are considered to be used. One of these alternatives is Reversible Computing (RC) [6], which can provide ultra-low power computational circuits.

In this paper, we propose the joint usage of these two unconventional computing approaches, Residue Number System and Reversible Computing, to achieve ultra-efficient computing paradigm for the emerging applications. The ability of RNS to perform highly parallel and carry-free arithmetic is well suited for taking advantage of the features of reversible circuits. In other words, reversible logic can be efficiently used to implement RNS circuits. However, since all the available RNS structures are designed for ASIC implementation, rethinking of RNS architectures should be performed to adapt them to the properties of reversible circuits. The fundamental part of RNS systems is modular addition, since all parts of RNS including forward and reverse conversion are based on modular additions. Hence, the first step to implement RNS systems based on reversible circuits requires the design of efficient modular adders using reversible logic gates. This paper presents the first implementation of modulo 2^n-1 adders based on reversible gates. For these modular adders, which are frequently used in RNS structures, parallel-prefix and ripple-carry architectures are considered.

II. RESIDUE NUMBER SYSTEM ARCHITECTURE

The first step to architect a RNS is to select a moduli set according to the target application constraints and requirements. The moduli set consists of pair-wise relatively prime numbers $\{m_1, m_2, \ldots, m_n\}$, being the dynamic range the sequence of integers that can be uniquely represented in RNS, i.e. [0, M-1] with $M=m_1 \times m_2 \times \ldots \times m_n$ [4]. In order to decrease the complexity of hardware realization of RNS-based arithmetic, usually near power-of-two moduli are adopted, such as 2^n-1, 2^n and 2^n+1. Among these moduli, the simplest one to deal with is the 2^n, which does not require any specific modular arithmetic, jut the circuits for binary arithmetic. Apart from that, the most frequent co-prime number in moduli sets for RNS is 2^n-1, since moduli 2^n+1 is more complex and its representation requires on

This work was partially supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) under project UID/CEC/50021/2013 and the European FutureTPM project that has received funding from the European Union's Horizon 2020 under grant agreement No 779391.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

additional bit. Typical RNS moduli sets are $\{2^n\text{-}1, 2^{n+k}, 2^n\text{+}1\}$ [7], $\{2^n\text{-}1, 2^n, 2^{n+1}\text{-}1\}$ [8], $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{2n+1}\text{-}1\}$ [9], $\{2^k, 2^n\text{-}1, 2^n\text{+}1, 2^{n+1}\text{-}1\}$ [10] and $\{2^n\text{-}1, 2^n, 2^n\text{+}1, 2^{n+1}\text{-}1, 2^{n-1}\text{-}1\}$ [11]. The main arithmetic blocks of RNS are the forward converter, the modular arithmetic in the channels, and the reverse converter [12]. The forward converter translates the weighted binary number (X) to the residues (x_i's), according to the moduli, as:

$$X \xrightarrow{Forward\ Conversion} (x_1, x_2, \ldots, x_n) \quad (1)$$

Where

$$x_i = X \bmod m_i = |X|_{m_i} \quad for\ i = 1 \ldots n \quad (2)$$

Note that *mod* indicates the remainder of de integer division of X by m_i. Then, considering two numbers A and B as follows:

$$A = (a_1, a_2, \ldots, a_n) \quad (3)$$

$$B = (b_1, b_2, \ldots, b_n) \quad (4)$$

modulo arithmetic operations can be performed on residues as follows:

$$S = A \bullet B = (s_1, s_2, \ldots, s_n) \quad (5)$$

where

$$s_i = |a_i \bullet b_i|_{m_i}, \bullet \in \{+, -, \times\} \quad (6)$$

Finally, a reverse converter maps the results in the RNS domain to the regular weighted representation, by using, for example, the Chinese remainder theorem (CRT) [12]. Other RNS operations such as sign detection, magnitude comparison and overflow handling are optional, according to the target application, and harder to perform in the RNS domain. It should be mentioned that general division cannot directly be performed in RNS, but division by a constant, one of the moduli of the set, i.e. scaling, is easier to perform [13].

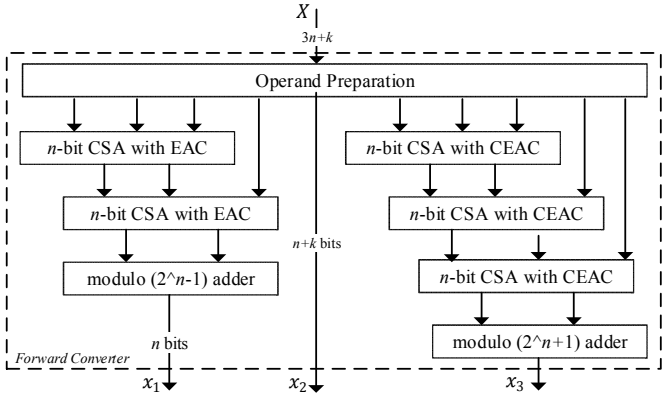

Fig. 1. The forward converter for the moduli set $\{2^n\text{-}1, 2^{n+k}, 2^n\text{+}1\}$ [7].

Most of the mentioned RNS operations are implemented using 3-to-2 carry-save adders (CSAs) with end-around carries (EACs) and 2-to-1 modular adders [14]. A full hardware design of RNS with moduli set $\{2^n\text{-}1, 2^{n+k}, 2^n\text{+}1\}$ is reported in [7], and herein forward and reverse converters for this moduli set are depicted in Figs. 1 and 2, respectively. It can be observed that CSAs and carry-propagate modulo 2^n-1 adders are the components required to implement a full RNS architecture, since arithmetic in a channel also reuires modulo adders and multipliers. Thus, to have an efficient modular adder is fundamental for RNS-based applications.

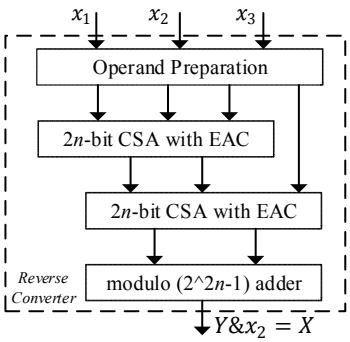

Fig. 2. The full reverse converter for the moduli set $\{2^n\text{-}1, 2^{n+k}, 2^n\text{+}1\}$ [7].

The CSA with EAC consists of independent full adders (FAs) which just combine the three inputs into two carry-save output vectors, as shown in Fig. 3. Its delay is just the delay of a single FA, while the overall area linearly depends on the width of the operands [14].

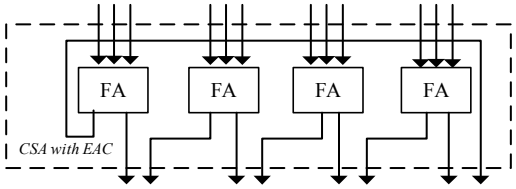

Fig. 3. The 4-bit CSA with EAC structure [14].

Note that CSA with CEAC is similar to CSA with EAC, just the end-around carry is complemented. Modular carry-propagate adders can be designed based on different architectures, from low-cost ripple-carry adders (RCAs) [15] (Fig. 4) to fast parallel-prefix adders (PPAs) [16]. The PPAs architectures can provide a good trade-off between circuit's parameters, being popular in RNS arithmetic circuits [17, 18].

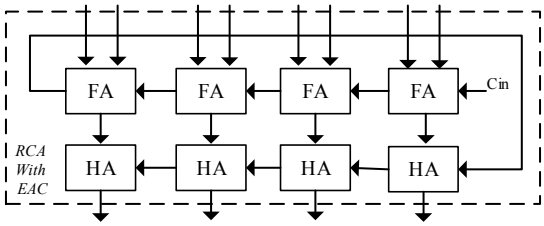

Fig. 4. The 4-bit modulo 2^n-1 adder based on ripple-carry method, namely RCA with EAC [15].

III. REVERSIBLE GATES

Reversible circuits provide a one-to-one relation between inputs and outputs, therefore inputs can be recovered from outputs. This interesting feature results in significant power-saving in digital circuits [20]. Classical digital gates are not reversible, reversible gates should be designed as basic components to design logical reversible circuits. Well known reversible gates are Feynman, Peres and HNG [20, 21]. The block diagrams of these gates are presented in Fig. 5.

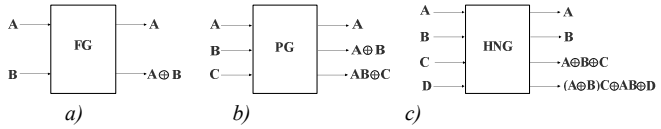

Fig. 5. The reversible gates: a) Feynman; b) Peres and c) HNG.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

The Feynman or controlled not (CNOT) gate is frequently used in reversible circuits, since it can provide exclusive OR (XOR) as well as copy and complement of the input. Since reversible circuits do not take advantage of fan-out, this gate can be used to achieve two copies of the same input by setting the other input of the gate to the zero-logic level. Similarly, by setting the second input of the CNOT to one-logic level, we can achieve the complement of the other input [20].

IV. Modular Adder Design Using Reversible Circuits

This section presents the reversible implementation of three modular adder structures that are frequently applied to RNS.

A. The CSA with EAC

The CSA is a 3-to-2 compression unit that is very popular for regular arithmetic as well as in RNS architectures due to its speed and cost. According to Fig. 3, a CSA can be built by using n FAs for adding three n-bit operands. According to [21], the HNG reversible gate can be used to realize a FA by setting the fourth input of HNG to the zero-logic level, as shown in Fig. 6.

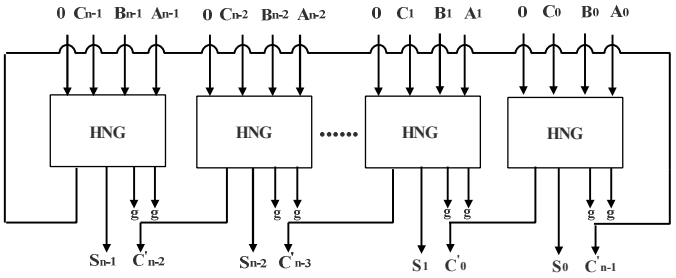

Fig. 6. The CSA with EAC using HNG reversible gates.

The quantum depth and cost of a HNG gate is 5Δ and 6, respectively [22]. Therefore, the total quantum depth and cost of a n-bit CSA with EAC will be 5Δ and 6n, respectively, since the delay of a CSA is equal to the delay of just one FA. Besides, the final reversible circuits will have n constant inputs and 2n garbage outputs.

B. The RCA-based Modulo Adder

As shown in Fig. 4, the RCA with EAC for modulo 2^n-1 addition of two n-bit numbers, requires n FAs and n HAs in the first and second levels, respectively. Similar to CSA, FAs can be realized with HNG gates. Besides, the Peres reversible gate can be used to implement a HA, where the third input bit is set to zero, as shown in Fig. 7. The final quantum cost of the RCA with EAC for two n-bit operands is 6n+4n=10n, since the individual quantum cost and depth of a Peres gate is 4. Besides, the total quantum depth of the RCA with EAC is ((3×(n-1)+4+(3×(n-1)+5))Δ. Furthermore, the total constant inputs and garbage outputs are 2n and 3n, respectively, since one of the inputs of HNG and Peres gates is zero, and also two and one outputs of HNG and Peres gates, respectively, are not used.

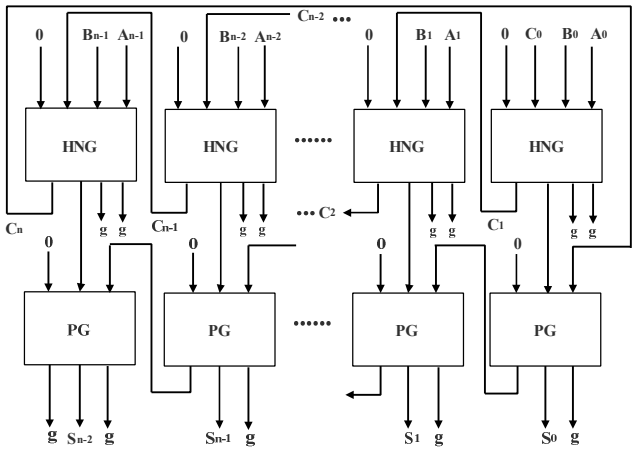

Fig. 7. The RCA with EAC using HNG and Peres reversible gates.

C. The PPA-based Modulo Adder

The heart of any PPA adder is a carry-computation network which consists of black and gray cells, as depicted in Fig. 8. There are different carry-computation networks which can be used for designing PPAs, as illustrated in Fig. 8 [19].

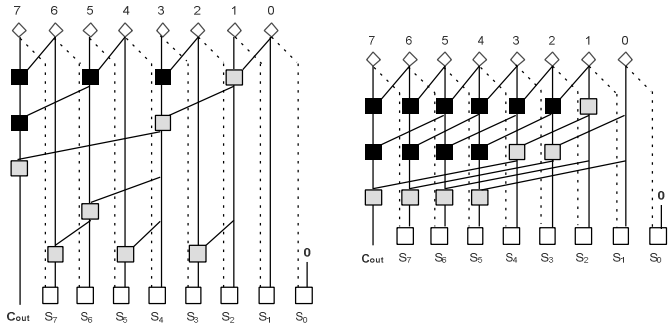

Fig. 8. The regular parallel-prefix adder structure based on (*left side*) Brent-Kung and (*right side*) Kogge-Stonemethods [19].

The reversible implementation of different PPAs are presented in [22], and it is shown that the Brent-Kung adder has the least quantum cost among the prefix structures. Due to this, the Brent-Kung adder is also selected herein as the basis to design modulo 2^n-1 adder with reversible gates. The Zimmerman's method [16] can be used to transform the regular Brent-Kung adder into a modulo 2^n-1 adder, by inserting a row of black cells to ad the carry-out, i.e. the end-around carrys as shown in Fig. 9.

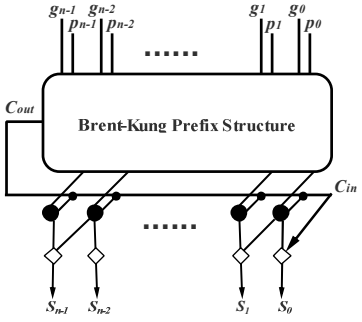

Fig. 9. The modulo 2^n-1 Brent-Kung prefix adder architecture [15].

The modulo 2^n-1 PPA consists of three main parts: generate and propagate signal preparation, prefix carry-computation network and post processing to produce final carry and sum bits. The reversible implementation of the first and the second parts have been proposed in [22]. Herein, we also use the reversible implementation of Brent-Kung carry-computation network as follows. First, generate and propagate signals should be computed using operands bits as:

$$p_i = a_i \oplus b_i \quad (7)$$
$$g_i = a_i b_i \quad (8)$$

The (7) and (8) can be simply implemented using a Peres gate with the third input reset to zero. The carry-computation network involves black and gray cells in Fig. 8, which perform the following operations to achieve group propagate and generate signals [22]:

$$P_{[i:j]} = P_{[i:k]} P_{[k-1:j]} \quad (9)$$
$$G_{[i:j]} = G_{[i:k]} + P_{[i:k]} G_{[k-1:j]} = G_{[i:k]} \oplus P_{[i:k]} G_{[k-1:j]} \quad (10)$$

The black cell requires two Peres gates to produce $G_{[i:j]}$ and $P_{[i:j]}$. The fan-out gate is also considered as a part of black cell to repeat $P_{[i:k]}$. Besides, the gray cell just needs one Peres gate to produce $G_{[i:0]}$, as shown in Fig. 10.

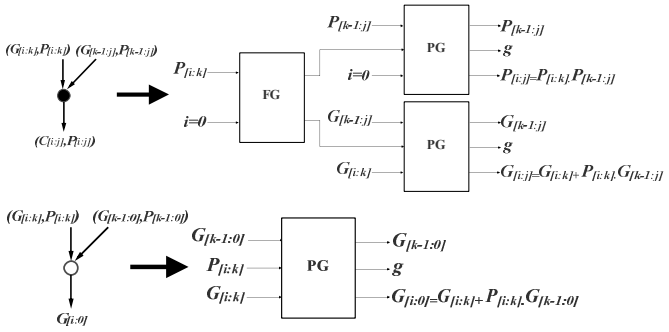

Fig. 10. The internal prefix cells implelemtation using Peres gates [22].

Finally, as shown in Fig. 9, the Post processing part includes a level of black cells to apply the C_{out} , i.e. EAC, to the middle carries, followed by sum cells to produce the sum. The black cells in this last level are different from the internal cells, since they have three inputs as depicted in Fig. 11. It can be seen that Peres gates are efficiently used to realize that cell. The sum cells can be simply implemented using a Feynman gate, since just a XOR is needed to produce each sum.

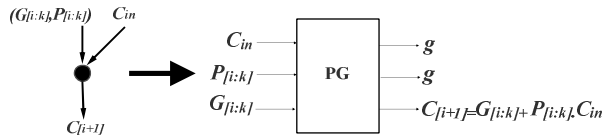

Fig. 11. The black cell used in the EAC level of the prefix modular adder.

The total quantum cost and depth as well as number of constant inputs and garbage outputs for the regular Brent-Kung adder is calculated in [22]. Therefore, it is just necessary to introduce one level of black cells to [22] to derive the quantum depth and quantum value for the proposed modular adder.

V. Performance Comparison

The total quantum cost and depth as well as number of constant inputs and garbage outputs of the different adders are presented in Table I. It can be seen that, as it was expected, the proposed 2^n-1 modulo adders have higher cost and depth than the equivalent binary adders. The proposed prefix-based modular adders have 19.81% and 7.55% overhead in terms of cost and depth, respectively, in comparison to the regular Brent-Kung prefix adder for n=32. However, for the same width, RCA with EAC results in 40% and 49.2% overhead in cost and depth, respectively, in comparison to the regular RCA. Therefore, it can be concluded that designing prefix-based modular adder results in less overhead than RCA-based design, in reversible logic. Besides, it can be seen from Table I that a 3-to-2 compression unit, like CSA, which is quite frequently used in RNS circuits, can be implemented quite efficiently using reversible gates.

VI. Conclusions

This work presents the reversible design of modular adders, a basic and fundamental element of RNS-based architectures. It is shown that a modulo 2^n-1 parallel-prefix adder can be designed using small overheads over regular prefix adders. The next steps, which should be considered for future work, are re-formulating the RNS operations, such as reverse conversion, sign detection and scaling, to adapt them to be implemented with reversible gates. They can benefit from the efficient proposed reversible-based modular adders. It is expected that this paper opens a new and substantial field of research to join modular arithmetic and reversible computing, resulting in efficient computational architectures for the post-Moore era.

TABLE I. Performance Comparison of Adders based on Reversible Logic

Type	Adders	Circuit Parameters															
		Quantum Cost				Quantum Depth (Δ)				Constant Inputs				Garbage Outputs			
		8	16	32	64	8	16	32	64	8	16	32	64	8	16	32	64
Regular [21]	Kogge-Stone	158	446	1166	2894	19	24	29	34	28	84	228	580	43	115	291	707
	Brent-Kung	104	239	518	1085	29	39	49	59	16	38	84	178	31	69	147	305
	Sklansky	196	496	1200	2816	20	25	30	35	32	80	192	448	52	128	304	704
	RCA	48	96	192	384	27	51	99	195	8	16	32	64	16	32	64	128
Modular Proposed	Brent-Kung w. EAC	136	303	646	1341	33	43	53	63	16	38	84	178	47	101	211	433
	CSA with EAC	48	96	192	384	5	5	5	5	8	16	32	64	16	32	64	128
	RCA with EAC	80	160	320	640	51	99	195	387	16	32	64	128	25	49	97	193

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

REFERENCES

[1] T.M. Conte, E.P. DeBenedictis, P.A. Gargini, and E. Track, "Rebooting Computing: The Road Ahead," *Computer*, vol. 50, no. 1, pp. 20-29, 2017.

[2] M. Alioto (Ed.), *Enabling the Internet of Things: From Integrated Circuits to Integrated Systems*, Springer , 2017.

[3] A.S.Molahosseini, L.Sousa and C.H. Chang (Eds.), *Embedded Systems Design with Special Arithmetic and Number Systems*,Springer, 2017.

[4] C.H. Chang, A.S.Molahosseini, A.A.Emrani Zarandi, and T.F.Tay, "Residue Number Systems: A New Paradigm to Datapath Optimization for Low-Power and High-Performance Digital Signal Processing Applications," *IEEE Circuits and Systems Magazine*, vol. 15, no. 4, pp. 26-44, 2015.

[5] L. Sousa, S. Antão, and P. Martins, "Combining Residue Arithmetic to Design Efficient Cryptographic Circuits and Systems," *IEEE Circuits and Systems Magazine*, vol. 16, no. 4, pp. 6-32, 2016.

[6] E.P. DeBenedictis, J.K. Mee, and M.P. Frank, "The Opportunities and Controversies of Reversible Computing," *Computer*, vol. 50, no. 6, pp. 76-80, 2017.

[7] R. Chaves and L. Sousa, "Improving RNS multiplication with more balanced moduli sets and enhanced modular arithmetic structures,"*IET Computers & Digital Techniques*, vol. 1, n. 5, pp. 472-480, 2007.

[8] A. Hiasat, "An Efficient Reverse Converter for the Three-Moduli Set (2^{n+1}-1, 2^n, 2^n-1)," *IEEE Transactions on Circuits and Systems-II*, vol. 64, no. 8, 2017.

[9] A.S. Molahosseini, K. Navi, C. Dadkhah, O. Kavehei, S. Timarchi, "Efficient reverse converter designs for the new 4-moduli sets {2^n–1, 2^n, 2^n+1, 2^{2n+1}–1} and {2^n–1, 2^n+1, 2^{2n}, 2^{2n}+1} based on new CRTs," *IEEE Transactions on Circuits and Systems-I,* vol. 57, no. 4, pp. 823-835, 2010.

[10] P. Patronik and S.J. Piestrak, "Design of Reverse Converters for General RNS Moduli Sets {2^k, 2^n-1, 2^n+1, 2^{n+1}-1} and {2^k, 2^n-1, 2^n+1, 2^{n-1}-1} (*n* even)," *IEEE Transactions on Circuits and Systems-I,* vol. 61, no. 6, pp. 1687-1700, 2014.

[11] B. Cao, C. H. Chang and T. Srikanthan, "A Residue-to-Binary Converter for a New Five-Moduli Set," *IEEE Transaction on Circuits and Systems-I,* vol. 54, no. 5, pp. 1041-1049, 2007.

[12] P.V.A. Mohan, *Residue Number Systems: Theory and Applications*, Springer, 2016.

[13] A.S. Molahosseini, A.A.E. Zarandi, P. Martins, and L. Sousa, "A Multifunctional Unit for Designing Efficient RNS-based Datapaths," *IEEE Access*, accepted, 2017.

[14] S.J. Piestrak, "A high speed realization of a residue to binary converter," *IEEE Transactions on Circuits and Systems-II,* vol. 42, pp. 661-663, 1995.

[15] J.L. Beuchat, "Some modular adders and multipliers for field programmable gate arrays," In Proc. of*IEEE International Symposium on Parallel and Distributed Processing*, 2003.

[16] R. Zimmermann, "Efficient VLSI implementation of modulo 2^n±1 addition and multications," in Proc.of *IEEE Symposium on Computer Arithmetic (ARITH)*, 1999.

[17] R.A. Patel, M. Benaissa and S. Boussakta, "Fast Parallel-Prefix Architectures for Modulo 2^n-1 Addition with a Single Representation of Zero," *IEEE Trans. Computers*, vol. 56, no. 11, pp. 1484-1492, 2007.

[18] A.A.E. Zarandi, A.S. Molahosseini, M. Hosseinzadeh, S. Sorouri, S.F. Antão and L. Sousa, "Reverse Converter Design via Parallel-Prefix Adders: Novel Components, Methodology and Implementations," *IEEE Transactions on Very Large Scale Integration (VLSI) systems,* vol. 2, no. 374-378, p. 23, 2015.

[19] R. Zimmermann, ªBinary Adder Architectures for Cell-Based VLSI and Their Synthesis,º PhD Thesis, Swiss Federal Institute of Technology, Zurich, 1997.

[20] S.M.R. Taha, *Reversible Logic Synthesis Methodologies with Applicationto Quantum Computing*, Springer, 2016.

[21] M. Haghparast and K. Navi, "A Novel reversible BCD adder for nanotechnology based systems,". *American Journal of Applied Sciences*, vol. 5, no. 3, pp. 282-288, 2008.

[22] C. Vudadha, and et. al., "Design and Analysis of Reversible Ripple, Prefix and Prefix-Ripple Hybrid Adders," In Proc. of *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2012.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE