Some Problematic Areas of Redundant Number System: Current State and Future Directions

M S Chakraborty

Assistant Professor
Department of Computer Science
Indas Mahavidyalaya, Indas, Bankura, India

Abstract: Ability to address the standard bus/peripheral devices, admissibility of low-power computations and efficient handling of cryptographic operations are some important factors to judge the computational strength of a number system for full-fledged arithmetic processor design. In this paper, on the basis of a survey over the said parameters, the computational strength of redundant number system is assessed.

Index Terms **- Computer Arithmetic, Redundant Number Systems, Peripheral Devices, Low-Power Computations, Cryptography**

I. INTRODUCTION

Ability to address the standard bus/peripheral devices, admissibility of low-power computations and efficient handling of cryptographic operations are some important parameters to assess the fitness of the number system for full-fledged, standalone, arithmetic processor design. In this paper, on the basis of aggregating and unifying different discrete reports available, the redundant number system (RDNS) [1] will be investigated with respect to the stated parameters.

Formal discussions on RDNS need to introduce the signed-digit number system (SDNS) at first. SDNS is a positional number system where values are expressed in terms of the signed-digits and zero, rather than involving the conventional unsigned digits and the range of the allowed digit-set (DS) is restricted by the radix of the number system [1]. In general, a radix-r SDNS is defined on the DS $\{\bar{\alpha}, \bar{\alpha}+1, .., \bar{1}, 0, 1, .., \beta-1, \beta\}$ where $\alpha \geq 0$, $\beta \geq 0$ and $\alpha + \beta + 1 > r$. SDNS which is defined on the DS $\{\bar{1}, 0, 1\}$ is called the binary signed-digit number system (BSDNS). In order to make the representation of a SDNS feasible at hardware level, every digit of its DS requires encoding as {0,1} strings. For BSDNS two encodings have been widely used: two's-complement encoding (TCE) and positive-negative encoding (PNE) [1]. In recent time another encoding, Tripathy's encoding (TE), has attracted the attention too [2]. For BSDNS digit-wise encodings in TCE, PNE and TE are shown in Table 1.

Table 1: Different Encodings for Binary Signed-Digits ([1]-[2])

Encoding Schemes	**Interpretation of Binary Signed-Digit ↓**		
	$\bar{1}$	**0**	**1**
TCE→	(1,1)	(0,0)	(0,1)
PNE→	(0,1)	(0,0) or (1,1)	(1,0)
TE→	(1,0)	(1,1)	(0,0)

A restricted form of BSDNS which is getting more popular now-a-days is canonical signed-digit number system (CSDNS). For expressing a value in CSDNS, its non-zero digits must be non-adjacent [1] and minimum in number.

The basic advantage of using SDNSs over the conventional number system (CNS) is to control the subsequent carry propagation during addition and complement-formation by cracking the carry propagation chain (CPC), either partially or fully [1]. The inherent redundancy of SDNSs is fundamentally credited for the cracking of the CPC in addition/subtraction and due to mandatorily carrying redundancy SDNS is also referred to as the RDNS.

Some other notable features of RDNS are: allowing faster multiplier design, regularity in circuit design, fault tolerance and online arithmetic ([1], [3]). Digital signal processing, image processing and cryptography are seemingly some broad areas where RDNS may find room for wider applications [1]. However, the possible involvement of larger chip area persists as a matter of concern for the future of redundant arithmetic.

In built-in form RDNS can not address the standard bus/peripheral devices and so the issue actually rests with the degree of efficiency in transforming RDNS into the conventional form, which is called the reverse conversion (RC). Recently in a survey paper the principles behind RC algorithms have been elaborated [4]. However, the RC algorithms proposed after 2010 have not been considered in this paper [4]. On the other hand, for both of low-power computations and cryptographic operations no full-fledged reports are available in the literature.

In this paper, for RDNS, separate concise reports will be presented on the current state of RC (2010 onwards), low-power computations and cryptographic operations. The reports will be followed by uncovering a list of open problems for future study. Accordingly the rest part of the article will be organized with four sections.

II. ADDRESSING PERIPHERALS/BUSES

As mentioned in the Introduction section, inability of RDNS to address peripherals/bus, obviously requires more efficient RC algorithms so that the CNS-output of RC may immediately take care of communications with peripherals/bus. However, it has been observed that the merit of RDNS may be lost due to obligations to perform RC that attracts considerably high area, delay, power and some other overheads [4]. Mathematically RC may be viewed as SDNS-to-CNS mapping or function, $\Psi(): F \rightarrow Z$ as shown in Fig. 1.

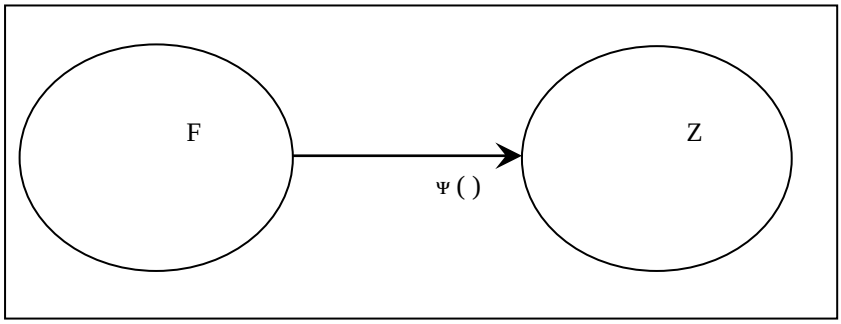

Fig. 1: Reverse Conversion as a Mapping

Consider radix = 2 as an ordinary case study. Here F denotes the set of all binary signed-digit numbers (BSDNs) and Z denotes the set of all two's-complement numbers. As F is redundant, two or more objects in F may have the same image in Z. For instance, $(\bar{1}1\bar{1}\bar{1})_F = (\bar{1}001)_F = (1001)_Z$. Obviously Ψ ( ) is not an injective mapping.

Now if possible assume that Ψ ( ) is not a surjective mapping too. It means ∃ member Z' in Z which is not the image of any object in F. Let $Z' = Z'_{n-1}Z'_{n-2}.....Z'_0$. It is well known that for such Z', ∃ $F' = F'_{n-1}F'_{n-2}.....F'_0$ in F where

$F'_0 = -Z'_0$, $F'_1 = Z'_0 - Z'_1$, $F'_2 = Z'_1 - Z'_2$, ... $F'_{n-1} = Z'_{n-1} - Z'_{n-2}$ [1].For example, for F' = 1001, Z' = $\bar{1}01\bar{1}$

It contradicts the assumption that Ψ ( ) is not a surjective mapping. In other words, Ψ ( ) is a non-injective and surjective mapping.

The RC problem has been being investigated for more than four decades for better resolution and covering all RC algorithms proposed till 2010, some unified study report is presented in article [4]. There are fundamentally two different approaches for performing the RC of any BSDN, $F=F_{n-1}F_{n-2}.....F_0$, into two's-complement number $Z=Z_nZ_{n-1}.....Z_0$. In one approach, say Approach-1, PNE is used in order to express $Z = F^+ - F^-$ and then $Z = F^+ + (-F^-)$ is determined employing some two's-complement adder [1] where $F^+ = F^+_{n-1}F^+_{n-2} \ldots . F^+_0$ and $F^- = F^-_{n-1}F^-_{n-2} \ldots . F^-_0$. In another approach, say Approach-2, for converting $\forall F_i$ the sign-information (SI) of the BSDN, $F_{i-1}F_{i-2} \ldots \ldots F_0$, say S_i, is prerequisite where $S_i = 1$ if $F_{i-1}F_{i-2} \ldots \ldots F_0 < 0$ and $S_i = 0$ otherwise. Subsequently $Z_i = F_i - S_i + r$ if $F_i - S_i < 0$ and $Z_i = F_i - S_i$ otherwise. Obviously the primitive versions of both Approach-1 and Approach-2 acquire linear time-complexity. Later faster, multi-level, reverse hierarchical (MLRH) versions of the Approach-1 and Approach-2, acquiring logarithmic time-complexity, have been introduced. In multi-level versions of both Approach-1 and Approach-2 computations are done in mainly two phases: firstly conversion control network generation (CCNG) and thereafter performing the block-wise conversion (BC) using the respective values produced by the conversion control network. In article [5] Approach-1 is followed and the CCNG and BC are performed by employing Kogge-Stone parallel prefix network [1] and carry-lookahead/ carry-select adders [1] respectively. In article [6] Approach-2 is followed and the CCNG and BC are performed in the form of carry-look-ahead network and equivalent bit conversion algorithm [7]. However, more recently article [8] and article [2] again used the linear version of Approach-2 employing the concept of inverted encoding of negabits [9] and TE respectively, outperforming the linear versions of some state of art algorithms for RC. In method [8] sign information is a pre-requisite for RC. However, for method [2] sign-information becomes available and operational implicitly.

For RDNS RC is closely related to signed-detection ([1], [4]). For any RDNS sign-detection is a complex operation that needs to locate the most-significant, non-zero digit of the number. Besides improving the performance of RC algorithms [4], superior sign-detection techniques may result in better performance of computer arithmetic algorithms for square rooting, division, branching, CORDIC and normalization of floating-point ([10], [11]) too. For faster sign-detection some hierarchical structures are commonly employed, like degree-4 reverse tree [12], carry-look-ahead network [13], multi-level reverse carry network [11]. The last recent sign-detection algorithm [13] employs an optimized reverse tree structure of degree-2 and it claims to outperform some state of art algorithms.

III. LOW-POWER COMPUTATIONS

There is an increasing trend towards designing low-power arithmetic circuits. Unless employing low-power arithmetic circuits, neither the increasing levels of integration nor the portability of handheld devices may be supported in longer run. However, high-speed and low-power often appear as two conflicting parameters. RDNS were originally developed for high-speed regardless of power implications. Although some other unconventional number systems, like LNS and RNS, have been investigated for low power arithmetic in details ([14], [15]), for RDNS few discrete reports are available, though some reports are too promising. In a full-fledged experimental study report it has been claimed that compared to two's-complement adder, redundant-adder might offer superiority not only in terms of energy-delay product, but also in terms of energy requirements [16]. A radix-16 redundant-multiplier is presented in [17] where it is shown that using the ordinary properties of redundant arithmetic the number of partial products can be substantially reduced and in addition, while accumulating the partial products the unnecessary inversions can be minimized. As a result the power consumption may be reduced to approximately half compared the traditional Booth multiplier and thus for redundant-arithmetic applications, the requirements for larger chip area seems to have been compensated by power savings. Energy efficient radix-16 sequential multiplier design is discussed in article [18]. The discussion on designing an energy-efficient 64-bit divider using RDNS is available in [19]. However, investigating the details of [19] is beyond the scope of this paper as it obviously involves floating-point computations. Constant-time addition using RDNS also has been exploited for designing energy-efficient processor [20] where some instruction of the traditional ISA or parts therein are proposed to be substituted or extended by redundant-binary addition(s). In this regard, besides the proposed redundant-division [19], special instructions are strived to be developed for three-operand addition, four-operand addition, multiplication and multiply-and-accumulate in redundant-binary platform as well as for the RC of redundant outputs of multiplication and division to two's-complement form. The code replacement algorithm presented in [20] feeds some intermediate representation like which is generated by typical compilers as input and produces the associated data flow graph (DFG). The DFG is then processed by a Branch-and-Bound sub-algorithm which considers the implications of all possible alternative solutions to energy consumptions and selects the best one in this regard.

IV. APPLICATIONS IN CRYPTOGRAPHY

Protection of data is becoming an increasingly important issue in line with the rapid growth of ICT. Employing cryptographic techniques is a common approach to protect data. RSA is a powerful and widely used cryptographic technique which involves complex arithmetic operations in the forms of exponentiation and modular (mod) computations. Although arithmetic algorithms for faster exponentiation have been known for quite a long time, no efficient algorithm for the mod-computing was available prior to 1990. At this point it was observed that using redundant-binary arithmetic a carry-free, non-restoring division algorithm might be developed [21]. As the mod-operation is nothing but an ordinary division which yields the remainder as result, the proposed carry-free division algorithm would be also supportive for designing a faster, single-chip RSA processor with less area and power requirements [21]. In the recent years ECC emerges as a potential contender to the RSA cryptosystem. Although the cryptographic strengths of both RSA and ECC may be equivalent, ECC uses smaller size keys for providing the same level of security. So even the computing systems with less memory and processing speed, including web servers, may be entitled to get higher data security and this is particularly important when the connectivity of devices have been rising day-by-day in the context of the Internet of Things (IoT). It has been found that the fast addition and multiplication enabling features of RDNS may also be used to achieve high-speed in elliptical curve cryptographic (ECC) processor too and the related design and implementation issues are discussed in [22]. However, from the perspective of ECC, employing CSDNS seems more promising compared to BSDNS [23].

V. CONCLUSION

In this paper, on the basis of a survey on the three parameters, namely, ability to address the standard bus/peripheral devices, admissibility of low-power computations and efficient handling of cryptographic operations, the computational strength of RDNS is assessed. The study uncovers a list of open problems in RDNS-platform for further investigations as shown in the Table 2:

Table 2: Future Directions

Sl No	Area	Problem
1	RC	Online Algorithm for RC of Un-Normalized Fractions
2	RC	The Comparative merits of method [5] and method [6] relative to each other
3	RC	Non-linear version of method [2]: Developing and Comparative Merit Analysis
4	RC	Non-linear version of method [8]: Developing and Comparative Merit Analysis
5	RC	Higher Radix RDNS [1]: Efficient RC
6	RC	Fault-Tolerant [1] Reverse Converter Design
7	Low Power Computing	Energy-Efficient Sign-Detection of RDNS
8	Low Power Computing	CORDIC functions [3]: Computing with Low-Power
9	Low-Power Computing	Energy-Efficient Method for Square Root Determination

Obviously the future scope of RDNS appears to be promising in line with the rapid growth of ICT. Employing cryptographic techniques is a common approach to protect data. RSA is a powerful and widely used cryptographic technique which involves complex arithmetic operations in the forms of exponentiation and modular (mod) computations. Although arithmetic algorithms for faster exponentiation have been known for quite a long time, no efficient algorithm for the mod-computing was available prior to 1990. At this point it was observed that using redundant-binary arithmetic a carry-free, non-restoring division algorithm might be developed [21]. As the mod-operation is nothing but an ordinary division which yields the remainder as result, the proposed carry-free division algorithm would be also supportive for designing a faster, single-chip RSA processor with less area and power requirements [21]. In the recent years ECC emerges as a potential contender to the RSA cryptosystem. Although the cryptographic strengths of both RSA and ECC may be equivalent, ECC uses smaller size keys for providing the same level of security. So even the computing systems with less memory and processing speed, including web servers, may be entitled to get higher data security and this is particularly important when the connectivity of devices have been rising day-by-day in the context of the Internet of Things (IoT). It has been found that the fast addition and multiplication enabling features of RDNS may also be used to achieve high-speed in elliptical curve cryptographic (ECC) processor too and the related design and implementation issues are discussed in [22]. However, from the perspective of ECC, employing CSDNS seems more promising compared to BSDNS [23].

REFERENCES

[1] Koren, *Computer Arithmetic Algorithms*, CRC Press, London: UK, 2001.
[2] S. S. Tripathy, R. K. Barik, M. Pradhan, *An Improved Conversion Circuit for Redundant Binary to Conventional Binary Representation*, in Proceedings of International Conference on Computational Intelligence, Communications and Business Analytics (CICBA), Kolkara: India, 2017, pp. 363–371.
[3] M. D. Ercegovac, T. Lang, *Digital Arithmetic*, Morgan Kaufmann Publishers (An Imprint of Elsevier), San Francisco: USA, 2004.
[4] M. S. Chakraborty, *Reverse Conversion Schemes for Signed-Digit Number Systems: A Survey*, Journal of Institution of Engineers (I): Series B, Vol. 97, 2016, pp. 589–593.
[5] Y. He, C.-H. Chang, *A Power-Delay Efficient Hybrid Carry-Lookahead/Carry-Select based Redundant Binary to Two's-Complement Converter*, IEEE Transactions on Circuits and Systems- I, Regular Papers, Vol. 55, 2008, pp. 336–346.
[6] S. K. Sahoo, A. Gupta, A. R. Asati, C. Shekhar, *A Novel Redundant Binary Number to Natural Binary Number Converter*, Journal of Signal Processing Systems, Vol. 59, 2010, pp. 297–307.
[7] Yun Kim, Bang-Sup Song, J. Grosspietsch and S. F. Gillig, "A carry-free 54b/spl times/54b multiplier using equivalent bit conversion algorithm," IEEE Journal of Solid-State Circuits, vol. 36, no. 10, pp. 1538-1545, 2001.
[8] R. K. Barik, M. Pradhan, R. Panda, *Efficient Conversion Technique from Redundant Binary to Non Redundant Binary Representation*, Journal of Circuits, Systems and Computers, Vol. 26, 2017, pp. 1750135 – 1–1750135–18.
[9] G. Jaberipur and B. Parhami, "Posibits, negabits, and their mixed use in efficient realization of arithmetic algorithms," 2010 15th CSI International Symposium on Computer Architecture and Digital Systems, 2010, pp. 3-9, doi: 10.1109/CADS.2010.5623646.
[10] J.E. Volder, "The CORDIC Trigonometric Computing Technique," IRE Trans. Electronic Computers, vol. 8, pp. 330-334, 1959.
[11] T. Lang and J.D. Bruguera, "Multilevel Reverse-Carry Computation for Comparison and for Sign and Overflow Detection in Addition," Proc. Int'l Conf. Computer Design (ICCD), 1999.

[12] T. Stouraitis, C. Chen, *Fast Digit-Parallel Conversion of Signed Digit into Conventional Representations*, Electronics Letters, Vol. 27, 1991, pp. 964–965.
[13] T. Srikanthan, S. K. Lam and Mishra Suman, "Area-time efficient sign detection technique for binary signed-digit number system," in IEEE Transactions on Computers, vol. 53, no. 1, pp. 69-72, 2004, doi: 10.1109/TC.2004.1255791.
[14] V. Paliouras and T. Stouraitis, "Low-power properties of the logarithmic number system," Proceedings 15th IEEE Symposium on Computer Arithmetic. ARITH-15 2001, 2001, pp. 229-236, doi: 10.1109/ARITH.2001.930124.
[15] V. Classon, Low Power Design using RNS, Ph.D. Thesis, 2014
[16] K. G. Smitha, A. H. Fahmy, A. P. Vinod, *Redundant Adders Consume Less Energy*, in Proceedings of IEEE APC on Circuits and Systems, Singapore, 2006, pp. 422–425.
[17] D. Crookes, M. Jiang, *Using Signed Digit Arithmetic for Low Power Multiplication*, Electronics Letters, 2007, pp. 13–14.
[18] S. Amanollahi and G. Jaberipur, "Fast Energy Efficient Radix-16 Sequential Multiplier," in IEEE Embedded Systems Letters, vol. 9, no. 3, pp. 73-76, 2017.
[19] S. Amanollahi and G. Jaberipur, "Energy-Efficient VLSI Realization of Binary64 Division With Redundant Number Systems" in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 3, pp. 954-961, 2017, doi: 10.1109/TVLSI.2016.2604346.
[20] S Amanollahi, G Jaberipur, Extended Redundant-Digit Instruction Set for Energy-Efficient Processors, ACM Transactions on Embedded Computing Systems (TECS) 17 (3), 2018, 1-21
[21] A. Vandemeulebroecke, E. Vanzieleghem, T. Denayer, P. G. A. Jespers, *A New Carry-Free Division Algorithm and Its Application to Single-Chip 1024-b RSA Processor*, IEEE Journal of Solid State Circuits, Vol. 25, 1990, pp. 748–755.
[22] H. Marzouqi, M. Al-Qutayri, K. Salah, D. Schinianakis and T. Stouraitis, "A High-Speed FPGA Implementation of an RSD-Based ECC Processor" in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 1, pp. 151-164, 2016, doi: 10.1109/TVLSI.2015.2391274.
[23] A. Sabu, "Comparitive study of RSD based and CSD based arithmetic modules of ECC" 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, pp. 867-870, doi: 10.1109/ICCONS.2017.8250587.