Fast Transformation of the Standard Output of Reverse Conversion of BSDNS: A Case Study

Dr. Madhu Sudan Chakraborty
Department of Computer Science
Indas Mahavidyalaya DT: Bankura, WB, India
E-mail:- *mailmschakraborty@rediffmail.com*
DOI: - https://doi.org/10.47531/MANTECH/ECC.2021.40

Abstract

Signed-Digit Number Systems have been resurging as a potential contender of the conventional radix-complement number system. However, even at the present level of technical advancement, the signed-digit number systems cannot cater to all computational needs. The ordinary signed-digit output of the common signed-digit arithmetic operations is to be transformed back into the conventional radix-complement form for further processing. This transformation is called the reverse conversion, and owing to consumption of significant delay, area and power overheads, it is often projected a major performance bottleneck of the signed-digit arithmetic. Even the subsequent radix-complement to sign-magnitude conversion, whenever needed, also attracts similar high overheads. Recently, Chakraborty and Mondal proposed a generic method for the constant-time conversion of the radix-complement output of signed-digit arithmetic operations into the sign-magnitude form with low overheads in order to counteract the high overheads of the reverse conversion as a whole. However, the Chakraborty-Mondal method has been proposed with abstract, generic terms. As the binary signed-digit number system has been being subjected to the most rigorous investigations among the various classes of signed-digit number systems on various issues, in this paper, the arithmetic constructs of the Chakraborty and Mondal algorithm is strived to be elaborated in sufficient details for the binary signed-digit number system, focusing its possible adaptation by the recent reverse conversion algorithm proposed by Sahoo, Gupta, Asati and Shekhar.

Keywords: *- Signed-digit number systems, Reverse conversion, Computer arithmetic, Constant-time conversion, Radix-complement to sign-magnitude transformation.*

INTRODUCTION

Signed-Digit Number Systems (SDNSs) [1] broadly refer to a collection of some classes of unconventional number systems, having the common configuration that in each class, every non-zero member of the digit-set (DS) contains the respective sign too. Any SDNS inherently carries redundancy, and the redundancy appears to be a key for performing constant-time addition/ subtraction [2]. As addition/ subtraction is the basic building block of any arithmetic unit (AU), the faster addition/ subtraction may further lead to speedy multiplication/ division for designing higher-speed AUs as a whole, too, compared to the conventional radix-complement (CRC) number system [2]. However, even at the present level of technical advancements, the SDNSs cannot cater for all computational needs ([2]-[3]), and obviously, the ordinary signed-digit (SD) output of the common signed-digit arithmetic (SDA) operations is to be transformed back into the CRC form for further processing. This transformation is called the reverse conversion (RC) ([2]-[3]). However, owing to the consumption of significant time, area and power overheads, RC have often projected a major performance bottleneck of the SDA [3]. Even besides the problematic issue of RC and non-compliance with the standard floating-point arithmetic, SDNSs are supposed to attract larger chip area and higher power consumption, too, owing to its inherent redundancy. As a result, the SDNSs seemed to have lost some advantages reported at the initial stage. However, with the reporting of some recent computational results in favour of SDNSs by outperforming the CRC number system over the issues of fault-tolerance [4], low-power arithmetic ([5]-[9]), floating-point computations ([10]-[11]) and cryptographic processor designing [12],

SDNSs have come back to the flashlight again, and the various issues of SDNSs have become more worthy for further investigations. In this regard, the particular class of SDNSs which has experienced the most rigorous investigations contains the DS $\{\bar{1},0,1\}$ and it is called binary signed-digit number system (BSDNS) [2]. In this paper, the conversion scheme proposed in [13] is strived to be customized for the BSDNS, focusing its compatibility with the recent reverse conversion algorithm (RCA) proposed in [14]. Actually following the problematic RC, even the subsequent radix-complement to sign-magnitude conversion, which is often needed, also attracts carry-propagation (CP) with high overheads. In this regard, the carry-free generic scheme recently proposed in [13] for converting the radix-complement output of SDA operations into the sign-magnitude form also involves low overheads to counteract the high overheads of the RC as a whole. However, the method [13] has been proposed at the abstract level only, and so at least a specific, realistic case study is obviously needed, which would be the major contribution of the paper.

The rest portion of the paper is organized into three sections. In the Preliminaries section, the relevant parts of the method [13] and RCA [14] are discussed in brief. In the section named Adaptation of Method [13]: by RCA [14], the required arithmetic equations (AEs) are realized. Finally, in the Conclusion section, the immediate outcomes, as well as some probable implications of the proposed work, are discussed.

THE PRELIMINARIES

A. The Reverse Conversion Algorithm Proposed in [14]

One of the RCAs for BSDNS published in recent years from some duly recognized platform is the Sahoo-Gupta-Asati-Shekhar algorithm [14]. The RCA [14] is based on the carry-look-ahead (CLA) interpretation of subtraction in order to realize the RC using the positive-negative encoding (PNE) [2]. In this regard, an 8-bit subtractor is shown in Fig. 1. In RCA [14] at first the binary signed-digit number (BSDN) input F = Fn-1Fn-2…....F0 is logically decomposed into adjacent, non-overlapping, 8-digit size blocks. For each block, the (G, P) pair, which actually represents the sign-zero information (SZI), is separately computed, and this information is passed to the suitable CLA network. For instance, a 64-digit BSDN-input 8-bit CLA network, as shown in Fig. 1, is appropriate. The CLA network yields the exact SZI for each block on the basis of the inter-dependence of blocks, and subsequently, the blocks may be reverse converted in parallel. However, at this stage, while reverse converting the blocks independently, the intra-block transformation is performed digit-serially, and in this regard, 2-bit equivalent bit conversion algorithm (EBCA) ([15]-[16]) shown in Table I, is employed.

Table I: 2-bit EBCA ([15]-[16])

Inputs		Outputs					
		When $D_{i-1} = 0$			*When $D_{i-1} = 1$*		
F_i	F_{i-1}	Z_i	Z_{i-1}	S_{i+1}	Z_i	Z_{i-}	S_{i+1}
$\bar{1}$	$\bar{1}$	0	1	1	0	0	1
$\bar{1}$	0	1	0	1	0	1	1
$\bar{1}$	1	1	1	1	1	0	1
0	$\bar{1}$	1	1	1	1	0	1
0	0	0	0	0	1	1	**1**
0	1	0	1	0	0	0	0
1	$\bar{1}$	0	1	0	0	0	0
1	0	1	0	0	0	1	0
1	1	1	1	0	1	0	0

It may be noted that in the article [14], negative, zero and positive signs are denoted as $\bar{1}$, 0 and 1, respectively, employing 2-bit SZI (Si, ZBi). In Fig. 1, there are three types of cells, X, Y and W. Regardless of the cell type, inward and outward arrows represent input and output to/ from the cell. For the borrow variable (D), initially $D_0 = 0$. An output-arrow started with a filled-circle indicates that the output is generated at that level, whereas an output-arrow started with a filled-rhombus indicates that the output is carry-forwarded from the next higher level to that level. The cell-X, cell-Y and cell-Z are shown in Fig. 2, Fig. 3 and Fig. 4 respectively in generic form, and accordingly, the AEs for cell-X, cell-Y and cell-Z can be shown as in (1), (2) and (3), respectively.

$$G_i = F_i^-, P_i = \overline{F_i^+} \quad (1)$$

$$\left.\begin{aligned} D_j &= G_{j-1,k} + P_{j-1,k}.D_k \\ G_{i,k} &= G_{i,j} + P_{i,j}.G_{j-1,k} \\ P_{i,k} &= P_{i,j}.P_{j-1,k} \end{aligned}\right\} \quad (2)$$

$$D_j = G_{j-1,k} + P_{j-1,k}.D_k \quad (3)$$

Thus F can be converted into its equivalent two's-complement (TC) form $Z = Z_{n-1}Z_{n-2}……..Z_0$.

Constant-Time Conversion of Two's Complement Output into Sign Magnitude Form [13]

In addition (3) is to be extended by (5) as follows:

$$\left.\begin{aligned} S_j &= G_{j-1,k} + P_{j-1,k}.S_k \\ ZB_j &= P_{j-1,k}.ZB_k \end{aligned}\right\} \quad (5)$$

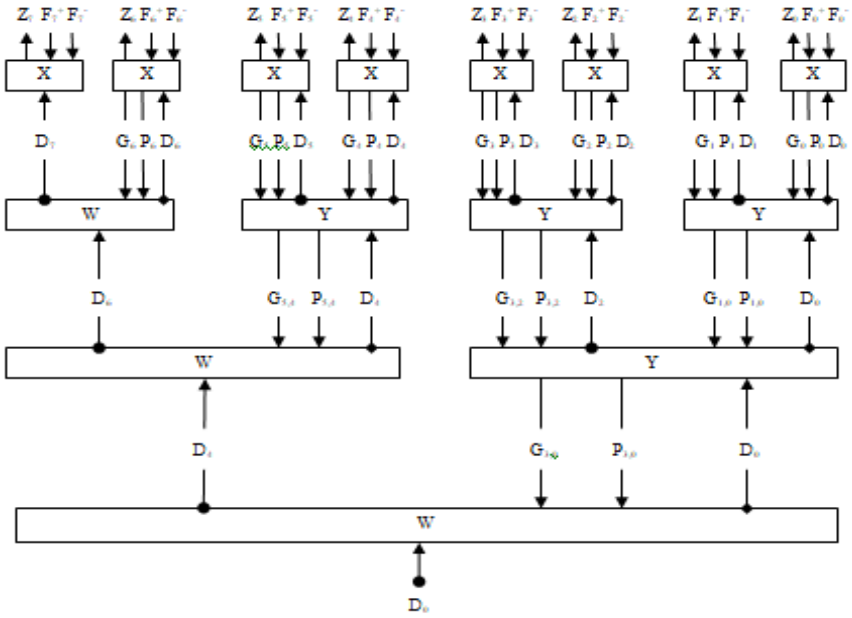

Fig. 1: CLA Realization of an 8-bit Subtractor

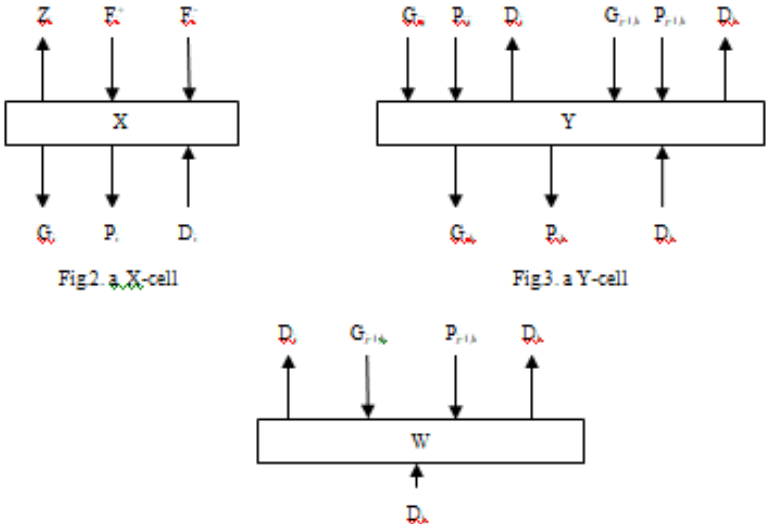

Fig.2. a X-cell

Fig.3. a Y-cell

Fig. 2: a W-cell

ADAPTATION OF THE METHOD [13] BY RCA [14]

For stage 2, the relationship of SZI and OBCT is shown in Table II.

Table II. Generating OBCT using SZI

Inputs			Output
Sign	SZI		OBCT (T_i)
	S_i	ZB_i	
$\bar{1}$	1	0	1
0	0	1	0
1	0	0	1

Simplification of the Table II yields:

$$T_i = \overline{ZB_i} \qquad (6)$$

For the rest the step 2 through step 4 of the Method [13] would be followed.

CONCLUSION

SDNSs are studied as an interdisciplinary topic of Electronics, Computer Science and Electrical Engineering for specific applications in digital signal processing, image processing and cryptography. Sahoo-Gupta-Asati-Shekhar is a recent time algorithm proposed for the RC of the BSDNS into the TC form. In this paper, it has been strived to show that the Sahoo-Gupta-Asati-Shekhar algorithm may be pre-processed at a constant time with low overheads to adapt Chakraborty-Mondal method. In other words, the TC-output of the Sahoo-Gupta-Asati- Shekhar algorithm may be efficiently converted into the sign-magnitude form (SMF) at constant time.

Thus the major contribution of this paper lies in demonstrating that for any SDNS-based AU, the performance degradation caused by inevitably

involving carry-prone, high-overhead RC may be compensated during the later stages by engaging carry-free, low-overhead, TC-to-SMF transformation of the RC-output. The future work of the author would be investigating the adaptation of the Chakraborty-Mondal method by the other RCAs.

REFERENCES

1. Avizienis, "Signed-digit number representations for fast parallel arithmetic", IRE Transactions on Electronic Computers, EC-10, pp. 389–400, 1961.
2. Koren, Computer Arithmetic Algorithms, CRC Press, London: UK, 2001.
3. M. S. Chakraborty, "Reverse conversion schemes for signed-digit number systems: a survey", Journal of Institution of Engineers (I): Series B, Vol. 97, pp. 589–593, 2016.
4. G. C. Cardarilli, S. Pontarelli, M. Re, A. Salsano, "Fault tolerant design of signed-digit based FIR filters", in Proceedings of IEEE International Symposium on Circuits and Systems, Greece, 2006, pp. 2809-2812.
5. S. Phatak, S. Kahle, H. Kim, J. Lue, "Hybrid signed digit representation for low power arithmetic circuits", in Proceedings of Low Power Workshop in Conjunction with ISCA, Barcelona: Spain, 1998.
6. Crookes, M. Jiang, "Using signed digit arithmetic for low power multiplication", Electronics Letters, vol 43, pp. 13–14, 2007.
7. 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. VIII. S. Amanollahi and G. Jaberipur, "Energy-efficient VLSI realization of binary64 division with redundant number systems," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 3, pp. 954-961, 2017.
8. Saba Amanollahi, Ghassem Jaberipur, "Extended redundant-digit instruction set for energy-efficient processors", ACM Transactions on Embedded Computing Systems, vol. 17, no. 3, pp. 1-21, 2018.
9. A.Kaivani, S. Ko, "Floating point butterfly architecture based on binary signed digit representation", IEEE Transactions on Very Large Scale Integration, vol. 24, pp. 1208–1211, 2016.
10. R. Watpade, P. Palsodkar, "BSD adder for floating point arithmetic: A review," in Proceedings of International Conference on Communication and Signal Processing (ICCSP), Chennai, 2017, pp. 0553-0556.
11. H. Marzouqi, M. Al-Qutayri, K. Salah, D. Schinianakis and T. Stouraitis, "A high-speed FPGA implementation of an RSD-based ECC processor", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 1, pp. 151-164, 2016.
12. M. S. Chakraborty and A. C. Mondal, "Reverse conversion of signed-digit number systems: transforming radix-complement output", Indonesian Journal of Electrical Engineering and Computer Science, vol. 4, pp. 665–669, 2016.
13. 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, pp. 297–307, 2010.
14. Y. Kim, B. -.S. Song, J. Grosspietsch and S. F. Gilling, "A carry-free 54×54 b multiplier using equivalent bit conversion algorithm, IEEE journal of Solid-State Circuits, vol. 36, no. 10, pp. 1538–1544, 2001.
15. W. Rulling, "A remark on carry-free binary multiplication," IEEE Journal of Solid-State Circuits, vol. 38, no. 1, pp. 159-160, Jan. 2003, doi: 10.1109/JSSC.2002.806267.