

Reverse Conversion of Signed – Digit Number System: Fast Transformation of Sign – Magnitude Output

M.S. Chakraborty

Dept. of Computer Science, Indas Mahavidyalaya, PO: Indas, Bankura (WB), India, PIN 722205

*Corresponding Author: mailmschakraborty@rediffmail.com, Tel.: (+91)8967936328

Available online at: www.ijcseonline.org

Accepted: 21/May/2018, Published: 21/May/20182018

Abstract— In spite of various advantages of using signed – digit number system, so far reverse conversion has remained a major performance bottleneck of signed – digit arithmetic. Yet, the sign – magnitude output algorithm(s) for reverse conversion of signed – digit number system is known to be extensible to provide radix – complement output at constant additional time. In other words, the additional delay due to the carry that may otherwise propagate once again, may be easily eliminated. This is supposed to nullify the negative aspects of reverse conversion on arithmetic units working on signed – digit number system in longer run. Thus significant performance enhancement of signed – digit number system(s) may be possible. However, as shown in this paper, the existing digit – parallel algorithm for sign – magnitude to radix – complement output transformation can not work correctly. So a modified scheme is proposed and its correctness is proved.

Keywords—Signed – Digit Number Systems, Reverse Conversion, Digit – Parallel Transformation, Conventional Form

I. INTRODUCTION

Signed – digit (SD) number (SDN) system (SDNS), introduced by Avizienis [1], is a widely – studied unconventional number system, defined on a set of digits that contains zero, positive and negative integers. At elementary level, the most important features of SDNS are its ability to support digit – parallel addition and complement – generation, which also act as a key to speed up some otherwise complex arithmetic operations [2]. This is what makes SDNSs suitable for applications over a wide area, from general purpose microprocessor to specialized areas like digital signal processing and cryptography. SDNSs also offer regularity in circuit design and seem to be suitable for hardware implementation [2]. In addition, recently, VLSI implementations of some arithmetic operations using SDNS have been found to consume lower power ([3], [4], [5]). However, due to associated area, delay and power overheads, the conversion of numbers from SD – form back to the conventional forms, known as the reverse conversion (RC), has remained a performance bottleneck of SD – arithmetic ([6], [7], [8]). RC seems to be essential as SDNSs may be applicable only at intermediate levels of computing ([2], [8]). Commonly, the output of RC appears in radix – complement form (RCF) [8]. Sometimes the RCF output needs to be converted to sign – magnitude form (SMF) for further internal processing or for user – interface. It is known that the RCF – output of typical RC algorithms can be converted

into SMF merely at constant additional time [9]. This is one approach to nullify to negative aspects of RC in longer run. The other approach holds for binary SDNS (BSDNS) only [10] and it is to integrate circuits for RC of BSDNS and two’s – complement to CSD recording into a single chip. On the other hand, sometimes the output of RC algorithm(s) also emerges in SMF and in this connection, a constant – time algorithm for SMF – RCF output transformation was claimed to have been developed in article [11]. However, article [11] states nothing about the correctness of the scheme proposed as its part. So in this article the correctness of the scheme will be checked and if invalidated, it will be modified by correcting the error – prone portion.

The rest portion of this paper is organized with six sections. In Section II the existing scheme is presented. In Section III the incorrectness of the existing scheme is uncovered. In Section IV some modification(s) of the existing scheme is proposed. In Section V the proposed modified scheme is proved to be correct. In Section VI the proposed scheme is sought to fit to typical RC scheme(s). In Section VII an example is given to clarify the operations of the proposed scheme (for typical RC algorithm(s)). Finally the study is concluded with Section 8, stating its implications for performance of SDNSs.

II. THE EXISTING WORK [11]

Let $X = X_{n-1}X_{n-2}\dots X_0$ ($n \geq 2$) be an ordinary radix r number represented in SMF. The existing conversion scheme to transform X into the equivalent RCF, say Y , with the same radix and equal no. of digits as $Y = Y_{n-1}Y_{n-2}\dots Y_0$ as in [11] is given below:

1. Status check bit (S_i) corresponding to each X_i is defined as:
 - 1.1. Initially: $S_0 = 0$
 - 1.2. For $0 \leq i \leq n-2$:
 - 1.2.1. If $X_i > 0$ then $S_{i+1} = 1$;
 - 1.2.2. Otherwise, $S_{i+1} = S_i$.
 - 1.3. Compute S_n as:
 - 1.3.1. If $X_{n-1} = 0$ then $S_n = 0$
 - 1.3.2. Otherwise, $S_n = 1$
2. For $0 \leq i \leq n-2$, Y_i digits are computed as:
 - 2.1.1. If $S_n = 0$ then $Y_i = X_i$
 - 2.1.2. If $S_n = 1$ then $Y_i = r - S_i - X_i$
3. Do:
 - 3.1. Set: $Y_{n-1} = X_{n-1}$
4. Stop

III. INCORRECTNESS OF EXISTING SCHEME [11]

Consider the conversion $(-4)_{10}$ from SMF to RCF in binary mode employing 5 bits. In this connection, the input 10100 is transformed to the output 11122, which is obviously an invalid representation of $(-4)_{10}$ in two's complement form. The example reveals that the existing scheme for SMF to RCF conversion [11] can not work correctly.

IV. PROPOSED MODIFIED SCHEME

1. Status check bit (S_i) corresponding to each X_i is defined as:
 - 1.1. Initially: $S_0 = 0$
 - 1.2. For $0 \leq i \leq n-2$:
 - 1.2.1. If $X_i > 0$ then $S_{i+1} = 1$;
 - 1.2.2. Otherwise, $S_{i+1} = S_i$.
 - 1.3. Compute S_n as:
 - 1.3.1. If $X_{n-1} = 0$ then $S_n = 0$
 - 1.3.2. Otherwise, $S_n = 1$
2. For $0 \leq i \leq n-2$, Y_i digits are computed as:
 - 2.1.1. If $S_n = 0$ then $Y_i = X_i$
 - 2.1.2. If $S_n = 1$ then
 - 2.1.2.1. If $(S_i = 0 \text{ and } X_i = 0)$ then $Y_i = X_i$
 - 2.1.2.2. Otherwise, $Y_i = r - S_i - X_i$
3. Do:
 - 3.1. Set: $Y_{n-1} = X_{n-1}$
4. Stop

V. PROOF OF CORRECTNESS

If $X \geq 0$ then $X_{n-1} = 0$; otherwise, $X_{n-1} = r - 1$. When $X \geq 0$, $Y = X_{n-1} X_{n-2} \dots X_0$ where $X_{n-1} = 0$. Obviously, the proposed conversion scheme works correctly.

When $X < 0$, $-X$ (> 0) can be represented as: $X_{n-1}|X|$ where $X_{n-1} = 0$ and $|X| = X_{n-2}X_{n-3}\dots X_0$ denotes the magnitude of X .

$$\text{Let } Z = X_{n-1} |X| + Y \quad (1)$$

For proving that the correctness of the proposed conversion scheme when $X < 0$, it is to be shown that:

$$Z = 0 \quad (2)$$

Considering all possible cases, the result of LSD – to – MSD digit – by – digit computations of Z as defined in (1) is shown in Table 1 $\forall i \in [0, n-2]$ using the following notations:

C_i = Carry – out from $(i-1)^{\text{th}}$ position in Z (or Carry – in to i^{th} position of Z)

Z_i = Digit at i^{th} position of Z

Initially $i = 0$, $C_0 = 0$.

Table 1 shows that $0 \leq Y_i \leq r-1 \forall i \in [0, n-2]$. As presented in table 1, computing for Z starts with either case 1 or case 2 and then may go through case 3. In this connection, APES graph is shown in Fig.1.

In the APES graph vertex V_j represents case no j as in Table 1 $\forall j \in [1, 3]$. Also in order to represent the flow of control more precisely without any loss of generality, vertex V_0 and vertex V_4 have been introduced to denote the unique start case and stop case respectively. Although nothing needs to be computed at V_0 ; V_4 has to compute Z_{n-1} and C_n (before the process terminates) where $X_{n-1} = 0$ and $Y_{n-1} = r-1$. Conditional transitions P1, P2; Q1, Q2 are defined on Figure 1 as below:

P1: Is $X_i = 0$?; P2: Is $X_i \neq 0$?; Q1: Is $i < n-1$?; Q2: Is $i \geq n-1$?

Transition without any level means it is unconditional. Any transition originated at vertex V_0 keeps i unchanged and all other transitions increase i by 1. Then computations as per table 1 is performed and the control is switched to the next vertex through the matching transition. APES graph as

shown in Fig. 1 clearly reveals that any execution sequence terminates with either $C_n = C_{n-1} = 0$ or $C_n = C_{n-1} = 1$. Therefore on dropping the end – around – carry for any valid input, it is seen that $Z_i = 0 \forall i \in [0, n - 1]$, regardless of the execution sequence. It means, even when $X < 0, Z = 0$. Obviously the proposed scheme works correctly. The correct transformation of two different representations for 0 in SMF into the unique 0 in RCF is ensured by step 3 of the scheme.

Table 1. Digit – Serial Computing for Z as defined in (1)

Case No	Inputs			Outputs			
	S_i	C_i	X_i	Y_i	S_{i+1}	$(Y_i + X_i) + C_i$	
						Z_i	C_{i+1}
1	0	0	0	0	0	0	0
2	0	0	N.Z	$r - X_i$	1	0	1
3	1	1	D/C	$r - 1 - X_i$	1	0	1

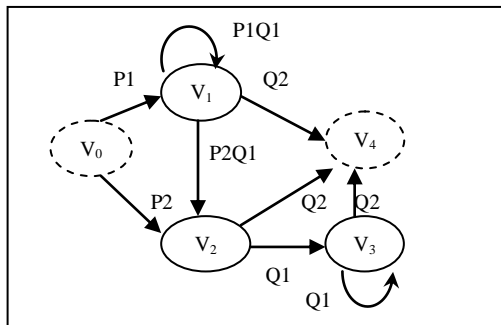


Figure 1. APES Graph Defined on Table 1

VI. IMPLICATIONS OF THE PROPOSED SCHEME IN THE CONTEXT OF TYPICAL RC OF SDNS

Let $D = D_{n-1}D_{n-2}.....D_0$ be a radix – r SDN defined on any valid digit set. Conversion control variable (T_i) corresponding to each D_i is defined as:

1. Initially: $T_0 = 0$
2. For $0 \leq i \leq n - 1$:
 - 2.1. If $D_i \neq 0$ then $T_{i+1} = 1$;
 - 2.2. Otherwise, $T_{i+1} = T_i$.

As expressed in section II let X be the equivalent SMF of D and signal S_{i+1} is defined corresponding to $X_i, \forall i \in [0, n - 1]$.

Corollary 1: $T_i = S_i \forall i \in [0, n - 1]$

If possible assume that $T_i \neq S_i$ for some $i \in [0, n - 1]$ and let j be the smallest value of i such that $T_j \neq S_j$. It means either $T_j = 0$ and $S_j = 1$ or $T_j = 1$ and $S_j = 0$.

Case 1: When $T_j = 0$ and $S_j = 1$

$T_j = 0$ means $T_{j-1} = 0, T_{j-2} = 0, \dots, T_0 = 0$

As j be the smallest value where T_j and S_j mismatches, $T_{j-1} = 0$ implies $S_{j-1} = 0$. It means a $j -$ digit partial signed – digit number (PSDN) as $D_{j-1}D_{j-2}.....D_0$ whose MSD is 0 and even all other digits are 0 is equivalent to a $j -$ digit (partial) signed – magnitude number as $X_{j-1}X_{j-2}.....X_0$ whose MSD is non – zero and all other digits are 0, which is a contradiction. Obviously, case 1 does not hold.

Case 2: When $T_j = 1$ and $S_j = 0$

Proceeding similar to case 1 it can be shown that case 2 does not hold.

Clearly $T_i = S_i \forall i \in [0, n - 1]$

Some algorithms for RC of SDNSs are based on direct or indirect sign – detection of PSDNs ([7], [8], [11]). In this connection, positive, negative and zero sign may be represented as 01, 11 and 00 respectively [12]. Clearly, for both positive and negative sign the least significant bits (LSBs) are 1 whereas for zero the LSB is 0. So the LSBs for signs of PSDNs may serve as T_i variables as presented in this section.

VII. EXAMPLE

Consider the output of RC of the binary SDN $D = \bar{1}010\bar{1}$ in SMF as $X = 11101$, which is to be further converted into binary RCF. Suppose that some RC scheme based on sign – detection of PSDNs has been employed that gives: $T_0 = 0, T_1 = \bar{1}, T_2 = \bar{1}, T_3 = 1, T_4 = 1, T_5 = \bar{1}$. Obviously $S_0 = 0, S_1 = 1, S_2 = 1, S_3 = 1, S_4 = 1, S_5 = 1$. Then SMF to TCF conversion in digit – parallel mode gives: $Y_0 = 1, Y_1 = 1, Y_2 = 0, Y_3 = 0, Y_4 = 1$. So $Y = 10011$ is the required binary TCF.

VIII. CONCLUSION

The conventional number system obviously needs no reverse conversion and clearly the question of carry – propagation does not arise. However, for conventional number system the transformation from SMF to RCF needs carry – propagation. On the other hand, though SDNSs need RC that necessarily involves carry propagation, it is known that by employing typical arithmetic algorithm for RC of SDNSs the output can be transformed from SMF to RCF without further carry propagation [11]. Thus the negative aspects of RC of SDNSs may be nullified in longer run. However, the existing SMF – to – RCF transformation scheme, presented as a part of [11], has been found to be incorrect. So in this paper some modification(s) has been proposed and the modified scheme has been proved to be correct.

REFERENCES

- [1] A. Avizienis, "Signed - digit number representation for fast parallel arithmetic", IRE Transactions on Electronic Computers, Vol. 10, Issue 3, pp.389 – 400, 1961.
- [2] I. Koren, *Computer Arithmetic Algorithms*, 2nd ed, University Press, Oxford, 2003.
- [3] G. Smitha, A. H. Fahmy, A. P. Vinod, "Redundant Adders Consume Less Energy", In Proceedings of IEEE APC on Circuits and Systems, Singapore, pp.422-425, 2006.
- [4] D. Crookes and M. Jiang, "Using signed - digit arithmetic for low power multiplication", Electronics Letters, Vol. 43, No. 11, pp. 613 – 614, 2007.
- [5] D.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, pp.1 – 7, 1998.
- [6] 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, Vol. 55, No.1, pp.336 – 346, 2008.
- [7] 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.
- [8] M.S. Chakraborty, "Reverse Conversion Schemes for Signed – digit Numbers Systems: A Survey", Journal of Institute of Engineers of India, Series B, Vol. 97, Issue 4, pp.589 – 593, 2016.
- [9] M.S. Chakraborty, A.C Mondal, "Reverse Conversion of Signed – Digit Number Systems: Transforming Radix – Complement Output", IJEECS, Vol. 4, No. 3, pp.665 – 669, 2016.
- [10] M.S. Chakraborty, S.K. Sao, A.C. Mondal, "Equivalence of Reverse Conversion of Binary Signed-Digit Number System and Two's-Complement to Canonical Signed-Digit Recording", IEEE International Conference on Recent Advancement of Information Technology (RAIT), IIT (ISM), Dhanbad, India, Vol. 2, pp.662 – 666, 2018.
- [11] T. Stouraitis, and C. Chen, "Fast Digit-Parallel Conversion of Signed-Digit into Conventional Representations", Electronics Letters, Vol. 27, Issue 11, pp.964-965, 1991.
- [12] T. Srikanthan, S.K. Lam, and M. Suman, "Area-Time Efficient Sign Detection Technique for Binary Signed-digit Number System", IEEE Transactions on Computers, Vol.53, Issue 1, pp.69-72, 2004.

Author's Profile

Mr. M S Chakraborty pursued Bachelor of Science with Hons in Mathematics from University of Burdwan, WB, India in 1998 and Master of Computer Applications from Regional Institute of Technology (presently N.I.T), Jamshedpur, Jharkhand, India in year 2002. He has been working as Assistant Professor in Department of Computer Science, Indas Mahavidyalaya, Bankura, WB, India, since 2007 and also pursuing Ph.D from Burdwan University, Burdwan, WB, India. He is a member of IEEE since 2011. He has published 6 research papers in reputed international journals (including SCOPUS – indexed) and conferences (including IEEE – sponsored) so far and most of his work is also available online. His main research work focuses on Computer Arithmetic and Theoretical Computer Science. He has more than 11 years of teaching experience and 4 years of Research Experience.

