JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.7, Special Issue.1, Jan 2019 E-ISSN: 2347-2693

An Unified Framework for Verifying Reverse Conversion Algorithms for Signed – Digit Number Systems

M.S. Chakraborty[1*], T. Ghosh[2], A.C. Mondal [3]

[1]Department of Computer Science, Indas Mahavidyalyaya, PO: Indas, DT: Bankura (WB), India
[2]Department of Computer Science, Bankura Samilani College, PO: Kenduadihi, DT: Bankura, India
[3]Department of Computer Science, Burdwan University, PO: Golapbagh, DT: Burdwan, India

[*]*Corresponding Author: mailmschakraborty@rediffmail.com*

Available online at: www.ijcseonline.org

Abstract — For signed–digit number systems, the proof of correctness of some reverse conversion algorithms remain ingrained in their developmental methods. As far as the remaining reverse conversion algorithms are concerned, either of no formal proof is given or different verification techniques are followed. Consequently at some points of time even some faulty algorithms were proposed through recognized platforms and sought to be projected as breakthroughs. In this paper, it is shown that a majority of available reverse conversion algorithms may be directly verified in generic form in a concise manner, merely using an ordinary mathematical induction technique, guided by the fundamental rules of typical reverse conversion. In addition to that, as evident from the case study which is presented in this paper, both the verification process and outcomes of reverse conversion, using the former class of algorithms, may be easily used to scrutinize the other more complex reverse conversion algorithms. Thus a unified framework for verifying reverse conversion algorithms for signed – digit number systems might have been obtained.

***Keywords*—**Reverse Conversion Algorithms, Signed – Digit Number Systems, Verification of Correctness, Unified Framework

I. Introduction

Delay – area – power efficient reverse converter design [1] is an open problem of signed – digit arithmetic [2]. Performing a detailed investigation of the existing reverse conversion algorithms from various quarters ([1] – [5]), particularly for their correctness, is seemingly a pre–condition for developing and validating even better reverse conversion algorithms. For the signed–digit number systems, the proof of correctness of some reverse conversion algorithms remain ingrained in their developmental methods ([6] – [14], [16]), whatever complex the procedure may be [14]. As far as the remaining reverse conversion algorithms are concerned, either of no formal proof is given ([15] – [17]) or different verification techniques have been followed ([18] - [21]). Consequently at some points of time even some faulty algorithms were proposed through recognized platforms and sought to be projected as silver – bullets ([15], [17]). This paper is an extension of the work [22] where the existing reverse conversion algorithms for binary signed – digit number system were visited category – wise and their proofs were sought to be generalized.

It may be recalled that the existing reverse conversion algorithms are known to be classified into three categories [5]:

Category – 1: Partial Signed – Digit Numbers' Sign – Detection based Algorithms ([6] – [12], [17] – [20])

Category – 2: One – Time Split, Add/ Subtract, Carry/ Borrow Manipulation based Algorithms [13]

Category – 3: Others ([14] – [16])

In the remaining part of this paper, it will be shown that reverse conversion algorithms belong to category – 1 and category – 2 may be directly verified in a concise manner, merely using an ordinary mathematical induction technique, guided by the fundamental conversion rules for category – 1. In addition to that, evident from the case study which will be presented in this paper, both the verification process and outcomes of reverse conversion, using the category – 1 algorithms, may be used to validate the other more complex reverse conversion algorithms of category 3 [14].

II. Verifying Category – 1 Algorithms in terms of Elementary Conversion Rules

The "elementary rules" for reverse conversion of a n – digit radix – r signed – digit number input, F = $F_{n-1}F_{n-2}$…..F_0 into (n + 1) – digit radix – r radix – complement form output, Z =

$S_nZ_{n-1}Z_{n-2}…..Z_0$ [2], using category – 1 algorithms, are given by table, where each is S_i is a conversion control variable and $S_0 = 0$.

Table 1.Elementary Conversion Rules for Category – 1 Algorithms [2]

F_i	S_i	Z_i	S_{i+1}
> 0	0	F_i	0
= 0	0	F_i	0
< 0	0	$r + F_i$	1
> 0	1	$F_i – 1$	0
= 0	1	r – 1	1
< 0	1	$r + F_i – 1$	1

It may be noted that the necessary and sufficient condition for equivalence of F and Z is given by:

$\Sigma F_jr^j = – S_nr^n + \Sigma Z_jr^j$ where j varies from 0 to n – 1 (1)

The operating principle of table 1 can be mathematically expressed as:

$Z_i = F_i – S_i$, if $(F_i – S_i)$ acquires non negative sign; $r + F_i – S_i$, otherwise (2)

$S_{i+1} = 0$, if $(F_i – S_i)$ acquires non negative sign; 1, otherwise (3)

The correctness of the binary version of the conversion scheme presented in table 1 was provided in the context of article [18]. Later in article [19] the proof has been extended to some high – radix signed – digit number system. However, the proof seems to be fairly complex. In this section an easy proof is given for the correctness of category – 1 reverse conversion algorithms, in generic form, with respect to the rules stated in table 1, using a simple mathematical induction technique.

Basis of Induction

When n = 2, for various signed – digit inputs, the corresponding radix – complement outputs are computed as per table 1 and presented in table 2.

Table 2.Computing 3 – digit Z for 2 – digit F

Inputs		**Outputs**			
F_1	F_0	S_2	Z_1	S_1	Z_0
< 0	< 0	1	$r + F_1 – 1$	1	$r + F_0$
= 0	< 0	1	r – 1	1	$r + F_0$
> 0	< 0	0	$F_1 – 1$	1	$r + F_0$
< 0	= 0	1	$r + F_1$	0	F_0
= 0	= 0	0	F_1	0	F_0
> 0	= 0	0	F_1	0	F_0
< 0	> 0	1	$r + F_1$	0	F_0
= 0	> 0	0	F_1	0	F_0
> 0	> 0	0	F_1	0	F_0

It is found that for all 2 – digit radix – r SD – inputs AE (1) is satisfied where AE stands for arithmetic equation.

Inductive Step

Suppose that the scheme presented in table 1 works correctly for any k – digit input. Let the k – digit input and the corresponding (k + 1) – digit output are given by F (k) = F_{k-1}……..F_0 and Z (k + 1) = S_kZ_{k-1}……Z_0 respectively.

Then (1) directly gives:

$\Sigma F_jr^j = – S_kr^k + \Sigma Z_jr^j$ where j varies from 0 to k – 1 (4)

Merging of (2) and (3) yields:

$Z_i = (F_i - S_i) + S_{i+1}.r$ (5)

Rearranging the terms and substituting i by k (5) can be expressed as:

$F_k = Z_k + S_k – S_{k+1}.r$ (6)

(4) + (6) × r^k gives:

$\Sigma F_jr^j + F_kr^k = – S_kr^k + \Sigma Z_jr^j + Z_kr^k + S_kr^k – S_{k+1}r^{k+1}$ where j varies from 0 to k – 1 (7)

It means,

$\Sigma F_jr^j = – S_{k+1}r^{k+1} + \Sigma Z_jr^j$ where j varies from 0 to k (8)

(8) clearly shows that if F(k) and Z(k + 1) agree on their values then F(k + 1) and Z(k + 2) must also agree on their values. Therefore it can be concluded that all algorithms belong to category – 1 ([6] – [12], [17] – [20]) works correctly.

III. VERIFYING CATEGORY – 2 ALGORITHMS USING THE CONVERSION RULES FOR CATEGORY – 1

So far no algorithm of category – 2 is known except [13]. Moreover algorithm [13] is applicable to binary signed – digit number system only. As apart from category – 1 algorithms, table 1 also represents the truth table of a full subtractor and a subtractor can be equivalently represented by a two’s – complement adder, algorithm [13] is validated immediately [22]. This strategy may be extended for verifying reverse conversion of higher – radix signed – digit number systems, assuming their availability for time being, based on the following observations:

Observation 1: If $F_i > 0$, in generic form it may be split as: $F_i + p$ and p, as $F_i = (F_i + p) – p$ where $0 \leq F_i, p, F_i + p \leq r – 1$

Observation 2: If $F_i = 0$, in generic form it may be split as: p and p, because, $F_i = p – p$ where $0 \leq F_i, p \leq r – 1$

Observation 3: If $F_i < 0$, in generic form it may be split as: p and $– F_i + p$, as $F_i = p – (– Fi + p)$ where $0 \leq F_i, p, – F_i + p \leq r – 1$

This clearly shows that in whatever manner, the digits of a higher – radix signed – digit number system be split, the positional sum – digit and borrow remain same and they observe table 1. Thus any generic algorithm of category 2 may also be verified immediately.

IV. VERIFYING CATEGORY – 3 ALGORITHMS WITH REFERENCE TO THE CONVERSION RULES FOR CATEGORY – 1: A CASE STUDY

The mapping – based version of algorithm [14], say Algorithm 1, is given below:

1. Convert each F_i as $Y_i = (r – 1) – F_i$. It results $Y = Y_{n-1}Y_{n-2}\ldots\ldots Y_0$ as a string of minimally redundant positive digits.
2. Decompose each Y_i into Y'_i and Y''_i such that $Y_i = Y'_i + Y''_i$ where $Y'_i = Y''_i = \{0, 1, 2, \ldots,r – 1\}$. One common decomposition of Y_i is $(Y_i, 0)$, except where $Y_i = r$. In the latter case Y_i is to be decomposed as $(Y_i – 1, 1)$.
   Then compute: $S = Y' + Y''$ using some conventional radix – r adder.
3. Convert each S_i into k – bit one’s – complement form except the MSD. The MSD is to be presented in (k + 1) bit one’s – complement form where $r = 2^k$.

Add r^n to the resulting number and Z is generated.

In this section, the correctness of Algorithm 1 is sought to be proved by showing that for all inputs its outputs agree with the outputs generated by category – 1 algorithms for the same radix, r and digit – set { , 0, 1, …, r – 1}. In this regard, algorithm [14] and any algorithm of category – 1 that directly realizes table 1, will make run in parallel and the outputs will be compared. As output is the one and only the matter of interest in this paper, without any loss of generality the digit – serial version of the concerned algorithms may be assumed. For the least – significant positions the actual output of table 1 and output of step 2 of algorithm 1 is shown in table 3 where $1^c = -1$.

Table 3. Comparison of Outputs at Least – Significant Position

Input (Digit)	Output of table 1		Output of Step 2 of algorithm 1	
	Z_i	**S_{i+1}**	**SM_{i+1}**	**C_{i+1}**
1^c	r – 1	1	0	1
0	0	0	r – 1	0
X (>1)	X	0	r – 1 – X	0

As for algorithm 1 the final output for all digits excluding the most – significant digit is obtained by taking the diminished – radix complement form of the respective step 2 values, using table 3 it may be immediately inferred that at least – significant position the outputs of algorithm 1 and a category – 1 algorithms must agree.

Next the comparisons of outputs at all higher – significant positions are presented in table 4.

Table 4. Comparison of Outputs at Higher – Significant Positions

Inputs		Output of table 1		Output of Step 2 of algorithm 1	
CCV	**Digit**	**Z_i**	**S_{i+1}**	**SM_{i+1}**	**C_{i+1}**
0	1^c	r – 1	1	0	1
0	0	0	0	r – 1	0
0	X (>1)	X	0	r – 1 – X	0
1	1^c	r – 2	1	1	1
1	0	r – 1	1	0	1
1	X (>1)	X – 1	0	r – X	0

Again as for algorithm 1 the final outputs for all digits excluding the most – significant digits are obtained by taking diminished – radix complement of the respective step 2 values, using table 4 it may be immediately inferred that at all intermediate positions the outputs of algorithm 1 and category 1 agree. Then consider the most – significant position. For algorithm 1 the output of step 2 is given by: C_nD_{n-1} where C_n is the carry (or conversion control variable) and D_{n-1} is the positional digit and it acquires a value $C_nr + D_{n-1.}$ In step 3 the value is transformed to $(2r – 1) – (C_nr + D_{n-1}) = (r – C_n.r) + ((r – 1) – D_{n-1})$ which appears as $(1 – C_n)((r – 1) – D_{n-1})$. The least – significant part i.e. $(r – 1) – D_{n-1}$ agrees with the output of table 1 but the most – significant part i.e. $1 – C_n$ does not. In addition as either $C_n = D_n = 0$ or $C_n = D_n = 1$, $1 – C_n$ and D_n are complementary to each other and so a correction bit 1 is to be added to the n^{th} position, as done towards the end of step 3 of algorithm 1.

V. CONCLUSION

Reverse Conversion of signed – digit number systems is still an open area for investigations ([1], [23]). A partially – generalized proof for correctness of reverse conversion algorithms for binary signed – digit number system, discussed in [22], is sought to be extended for signed – digit number systems, in this paper, in generic form. A large number of existing algorithms can be verified by the proposed framework in straightforward manner. The rests (more complex algorithms) may be easily checked by arranging parallel run with some of the aforesaid algorithms. The proposed unified framework may even be referred to develop and verify more efficient reverse conversion algorithms.

REFERENCES

[1] M. S. Chakraborty, "*Reverse Conversion Schemes for Signed - Digit Number Systems: A survey"*, Journal of Institute of Engineers (India): Series B, Vol. **97**, pp. **589-593**, **2016**.

[2] I. Koren, *"Computer Arithmetic Algorithms"*, CRC Press, **Oxford**, **2001**.

[3] 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"*, In the Proceedings of the 2018 IEEE International Conference on Recent Advancements in Information Technology, **IIT/ ISM, Dhanbad, India**, pp. **662 – 666**, **2018**.

[4] 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**.

[5] M. S. Chakraborty, A. C. Mondal, S. K. Sao, *"Towards Relating Some Methods of Signed – Digit Arithmetic"*, In the Proceedings of the 2018 International Conference on Mathematics, **St. Thomas College**, **Thrissur**, **India, pp**. **47 – 54**, **2018.**

[6] 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**, pp. **1750135-1-1750135-18**, **2017**.

[7] S. S. Tripathy, R. K. Barik, M. Pradhan, *"An Improved Conversion Circuit for Redundant Binary to Conventional Binary Representation"*, In the Proceedings of the 2017 International Conference on Computational Intelligence, Communications and Business Analytics (CICBA), **Kolkara**, **India**, pp. **363-371, 2017**.

[8] 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**.

[9] G. Wang and M. P. Tull, *"A new Redundant Binary Number to two's complement Number Converter"*, In the Proceedings of the 2004 IEEE Region 5 Conference, **Norman**, **USA**, pp. **141 – 143**, **2004**.

[10] G. A. Ruiz, *"4bit CLA-based conversion from redundant to binary representation for CMOS simple and multi-output implementations"*, Electronics Letters, Vol. **35**, No. **4**, pp. **281-283**, **1999**.

[11] S. Yen, C. Laih, C. Chen and J. Lee, *"An Efficient Redundant - Binary Number to Binary Number Converter"*, IEEE Journal of Solid-State Circuits, Vol. **27**, No. **1**, pp. **109-112**, **1992**.

[12] T. Stouraitis, C. Chen, *"Fast Digit - Parallel Conversion of Signed Digit into Conventional Representations"*, Electronics Letters, Vol. **27**, No. **11**, pp. **964 – 965**, **1991**.

[13] 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**, pp. **336-346**, **2008**.

[14] S.-H. Shieh, C-W Wu, *"Asymmetric High-Radix Signed-Digit Number Systems for Carry-Free Addition"*, Journal of Information Science and Engineering, Vol.**19**, pp. **1015-1039**, **2003**.

[15] I. Choo and R. G. Deshmukh, "A novel conversion scheme from a redundant binary number to two's complement binary number for parallel architectures", In the Proceedings of the 2001 IEEE **SoutheastCon, Clemson**, **USA**, pp. **196 – 207**, **2001**.

[16] S. Veeramachaneni, M. K. Krishna, L. Avinash, S. Reddy, M. B. Srinivas, *"High - speed redundant binary to binary converter using prefix networks"*, in the Proceedings of the 2007 IEEE International Symposium on Circuits and Systems, **Hyderabad, India**, pp. **3271 – 3274, 2007.**

[17] 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**.

[18] H. R. Srinivas, K. K. Parhi, *"High – Speed VLSI Arithmetic Processor Architectures using Hybrid Number Representation"*, Journal of VLSI Signal Processing, Vol. **4**, Issue **2 – 3**, pp. **177 – 198, 1992.**

[19] V. Charoensiri, A. Surarerks, *"On-the-fly Conversion from Signed-Digit Number System into Complement Representation",* In the Proceedings of the 2006 IEEE International Symposium on Communications and Information Technologies, **Bangkok**, **Thailand**, pp. **1056-1061**, **2006.**

[20] H. R. Srinivas and K. K. Parhi, *"A Fast VLSI Adder Architecture"*, IEEE Journal of Solid-State Circuits, Vol. **27**, No. **5**, pp. **761-767**, **1992**.

[21] W. Rülling, "A Remark on Carry - Free Binary Multiplication", IEEE Journal of Solid - State Circuits, Vol. **38**, No. **1**, pp. **159-160**, **2003**.

[22] M. S. Chakraborty, T. Ghosh, A. C. Mondal, *"Towards a General Proof for the Correctness of Various Reverse Conversion Algorithms for Binary Signed – Digit Number System"*, Orally Presented in International Seminar on Quality of Teaching – Learning in Higher Education in India: Concerns and Challenges, organized by the Education Department, Bankura University, **Bankura**, **WB**, **India, 2018**.

[23] M. S. Chakraborty, "*Reverse conversion of signed - digit number system: Fast transformation sign – magnitude output"*, International Journal of Computer Sciences and Enginering, Vol. **6**, Issue 5, pp. **454** – **457**, **2018**.

Authors Profile

Mr. M S Chakraborty, MCA (REC, Jamshedpur, 2002) has been working as an Assistant Professor in Department of Computer Sciences, Indas Mahavidyalya, (a.t. Bankura University), PO: Indas, DT: Bankura, WB, India, since 2007. He is a member of IEEE and his primary thrust area is computer artimetic. He has authored 06 articles at reputed national/ international journals and presented 03 papers at international conferences.

Mr T Ghosh, MCA (BU, 2002) has been working as an Assistant Professor in Department of Computer Sciences, Bankura Sammilani College, (a.t. Bankura University), PO: Kenduadihi, DT: Bankura, WB, India, since 2008. His primary thrust area is soft computing.

Dr A C Mondal, PhD (BU) has been working as a Professor at the Computer Science Department, Burdwan University, Burdwan, WB, India. His thrust areas are soft computing and cryptgraphy. He has authored more than 15 articles at reputed national/ international journals and presented more than 8 papers at international conferences. Also he has been serving as advisory coomittee member and reviewer of several national/ international journals.