Comments on "Area – Time Efficient Sign Detection Technique for Binary Signed-Digit Number System" proposed by Srikanthan, Lam and Suman

M. S. Chakraborty
Department of Computer Sc
Indas Mahavidyalya
Bankura, WB, India 722205

S. K. Sao
Department of Computer Sc
J. K. College
Purulia, WB, India 723101

ABSTRACT
It is shown that the sign detection technique for binary signed-digit number system reported by Srikanthan, Lam and Suman [1] cannot work correctly. The root cause of the problem is detected and some modifications are proposed. The proposed modifications are found to be significant.

General Terms
Unconventional Number Systems, Algorithm, VLSI

Keywords
Area-time efficient, binary signed-digit number system, reverse tree structure, sign detection technique

1. INTRODUCTION
For binary signed-digit (BSD) number system [2], [3], sign detection is a complex operation, which is needed to support applications employing branching, square root, division, CORDIC techniques, rounding and normalization of floating-point operations etc. An area-time efficient sign-detection technique for binary signed-digit (BSD) number system based on the optimized reverse tree structure was introduced in [1], where $\bar{1}$ (= -1), 0, 1 of the input BSD number $\boldsymbol{Z} = \boldsymbol{Z^s_{n-1}}\,\boldsymbol{Z^v_{n-1}} \ldots\ldots \boldsymbol{Z^s_1 Z^v_1 Z^s_0 Z^v_0}$ are represented as 11, 00, and 01 respectively. At level 1, the inputs are processed by the primary units and at all higher levels, the inputs are processed by the secondary units. A primary unit computes the sign flag and zero flag of the group obtained by combining the sign-value bits of two consecutive BSDs. A secondary unit computes the sign flag and zero flag of the group obtained by combining the sign-zero flags of two consecutive groups. In [1] on the basic of merely specifying some characteristics of the concerned operations the logic equations for primary unit and secondary unit are shown in (1) and (2) respectively.

$$f^s(1) = Z^s_i . Z^v_i + Z^s_{i-1} . Z^v_{i-1} . \overline{Z^v_i} \qquad (1)$$

$$f^z = \overline{Z^v_i}\;\overline{Z^v_{i-1}}$$

$$f^s(j+1) = f^s_i(j) + f^s_{i-1}(j) . f^z_i(j) \qquad (2)$$

$$f^z(j+1) = f^z_i(j) + f^z_{i-1}(j)$$

In the following it will be shown that some logic equation(s) for the technique presented in [1] is incorrect and some modification(s) will be proposed.

2. ON INCORRECTNESS OF THE LOGIC EQUATION(S)
The incorrectness of some logic equation(s) for the technique presented in [1] will be demonstrated with two examples:

Example 1: Let $\boldsymbol{Z = 1\bar{1}00}$. Then, encoded inputs at level 1 are: $z^s_3 = 0$, $z^v_3 = 1$, $z^s_2 = 1$, $z^v_2 = 1$, $z^s_1 = 0$, $z^v_1 = 0$, $z^s_0 = 0$, $z^v_0 = 0$. Outputs of the primary units at level 1 are: $f^s_1(1)=0, f^z_1(1)=0, f^s_0(1)=0, f^z_0(1)=1$. Outputs of the secondary units at level 2 are: $f^s_0(2)=0, f^z_0(2)=1$. This is the final sign-zero information given by [1]. But the correct value of the final zero flag is 0, because, the given number is positive.

Example 2: Let $\boldsymbol{Z = 1100\bar{1}0\bar{1}0}$. Proceeding similar to example 1, the final sign zero information given by [1] is: $f^s_0(3)=1, f^z_0(3)=1$. But, the correct value of both the final sign flag and zero flag are 0, because, the given number is positive.

It seems that (1) or (2) or both are incorrect. So, the correct logic equations for the primary unit and the secondary unit need to be revisited through the complete set of I/O specifications (truth tables). Then, (1) or (2) or both can be invalidated and replaced on the basis of comparative studies.

3. DETERMINATION OF CORRECT LOGIC EQUATIONS
Table 1: Computations performed by a primary unit

z^s_i	z^v_i	z^s_{i-1}	z^v_{i-1}	$f^s(1)$	$f^z(1)$
0	0	0	0	0	1
0	0	0	1	0	0
0	0	1	0	×	×
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	×	×
0	1	1	1	0	0
1	0	0	0	×	×
1	0	0	1	×	×
1	0	1	0	×	×
1	0	1	1	×	×
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	×	×
1	1	1	1	1	0

Table 2: Computation performed by a secondary unit

$f^s_i(j)$	$f^z_i(j)$	$f^s_{i-1}(j)$	$f^z_{i-1}(j)$	$f^s(j+1)$	$f^z(j+1)$
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	×	×

0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	×	×
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	×	×
1	1	0	0	×	×
1	1	0	1	×	×
1	1	1	0	×	×
1	1	1	1	×	×

The computations performed by the primary units and secondary units as shown in table 1 and table 2 respectively, where, × represents don't care condition.

K-map based simplification of I/O information presented in table 1 yields the logic equation for the primary unit as shown in (3).

$$f^s(1) = z_i^s + z_{i-1}^s . \overline{z_i^v}$$

$$f^z(1) = \overline{z_i^v} . \overline{z_{i-1}^v} \qquad (3)$$

K-map based simplification of I/O information presented in table 2 yields the logic equation for the secondary unit as shown in (4).

$$f^s(j+1) = f_i^s(j) + f_{i-1}^s(j) . f_i^z(j)$$
$$f^z(j+1) = f_i^z(j) . f_{i-1}^z(j) \qquad (4)$$

4. COMPARATIVE STUDY

Since Boolean expression $(f_i^z(j) + f_{i-1}^z(j))$ *and* $(f_i^z(j) . f_{i-1}^z(j))$ are not equivalent, (2) will not give the correct output as per I/O relationship expressed in (4). So, (2) is needed to be replaced by (4). Since the Boolean expressions $(z_i^s . z_i^v + z_{i-1}^s . z_{i-1}^v . \overline{z_i^v})$ and $(z_i^s + z_{i-1}^s . \overline{z_i^v})$ are equivalent, (1) will give the correct output as per I/O relationship expressed in (3). Yet, as expression (3) appears in a reduced form, a comparative study is needed to check whether the method employing (3) (and also (4) obviously) would give better performance than [1] or not. In this connection, transistor count can be used for chip area estimation. But as delay estimation merely on the basis of the number of logic gates on the critical path can be misleading due to the fact that delays are largely influenced by their circuit topology (fan-in) and loading (fan-out), the method of the logical effort (LE) [4] can be used as a shorthand for delay estimation, which has been attested as a more reliable technique by many researchers [5], [6]. For fair comparison both designs assume to have been implemented with NOT gates and 2-input NAND gates only. The gates optimized under the following conditions: Input size of 20μm, maximal allowable gate size on the circuit of 30μm and an identical load of 30μm-equivalent inverter. These conditions are set to get reasonable transistor sizes for layout and an acceptable load to the sign detector circuit. Delay time is normalized with respect to FO4 inverter delay. Area and delay performance of the existing method and the proposed method for different input sizes is shown in table 3 and scaled area-delay product against different input sizes is shown in Fig 1, where, the reciprocal of the area and delay product of the proposed method for input size = 4 is taken as the scaling factor

.

Table 3: Area and Delay Performance of the existing method [1] and proposed method for different input sizes

Number of inputs (BSD)	Existing Method [1]		Modified Method (proposed)	
	Delay	Transistor count	Delay	Transistor count
4	3.83	72	3.37	56
8	5.16	160	4.70	128
16	6.50	336	6.03	272
32	7.83	688	7.36	560
64	9.16	1392	8.69	1136

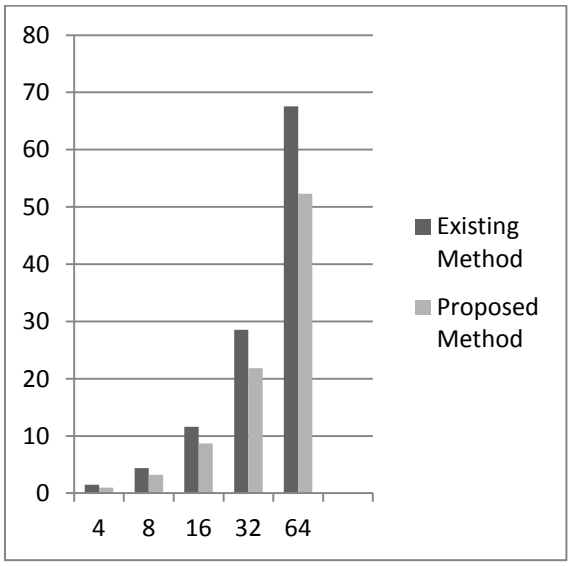

Fig 1: Scaled area-delay product against different input sizes (no. of BSDs).

5. CONCLUSION

An error has been detected in [1] and some modifications have been proposed. Although symbolically seem to be little, the proposed modifications can correct the reported error and have been found to improve the area – time performance of [1] significantly. Since, so far, no better sign – detection technique for BSD number system than [1] is available, the proposed modifications need to be considered for being applied in [1] so as to improve the performance of various applications of BSD number system requiring sign-detection and also all citations of [1] reported so far are required to be revisited. Although, in the recent year high-radix signed-digit number system [7], [8] and residue number system incorporating signed-digit architecture [9] has received considerable attention, no significant work has been reported so far for sign detection of these number systems. These problems will be studied as a part of future research work of the authors.

6. REFERENCES

[1] Srikanthan, T., Lam, S. K. and Suman, M. "Area–Time Efficient Sign Detection Technique for Binary Signed–digit Number System," IEEE Trans. Computers, vol. 53, no. 1, pp. 69-72, Jan. 2004.

[2] Avizienis, A., "Signed - digit number representation for fast parallel arithmetic", *IRE Transactions on Electronic Computers*, vol. EC - 10, no. 3, pp. 389 – 400, Sept 1961.

[3] Parhami, B., *Computer Arithmetic: Algorithms and Hardware Design*, 1st ed., Oxford Univ. Press, USA, Oct 2009.

[4] Sutherland, I., Sproull ,R. and Harris, D., Logical Effort: Designing Fast CMOS Circuits, 1st ed CA, USA: Morgan Kaufmann, Feb. 1999.

[5] Dao, H. Q. and Oklobdzija, V. G., "Application of Logical Effort on Delay Analysis of 64-Bit Static Carry-Lookahead Adder",35th Annual Asilomar Conference on Signals, Systems and Computers Pacific Grove, California, Nov. 4-7, 2001.

[6] He, Y. and Chang, C. – H., "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, Feb. 2008.

[7] Gorgin, S. and Jaberipur G., "A Family of High Radix Signed Digit Adders", in *20th IEEE Symposium on Computer Arithmetic*, Tubingen, Iran, July 2011.

[8] Naderpor, F., and Ko, S., "Improved Design of High - Radix Signed - Digit Adders," in *International Symposium on Electronic System Design*, Kolkata, India Dec. 2012.

[9] Wei S., "Residue-binary number conversion using signed-digit arithmetic for a four-moduli set", in *Proceedings of IEEE Region 10 Conference (TENCON) 2012*, Cebu, Philippines, pp. 1-4, Nov 2012.