A Division Algorithm for Signed-Digit Arithmetic

CHIN TUNG

Abstract—**The application of a fast division algorithm, particularly suitable for floating-point arithmetic, to signed-digit number systems is described. This method, based on the method of A. Svoboda, is performed in two steps: 1) the divisor is adjusted to be of the form $(1+e)$ where e is a fractional quantity, while the dividend is adjusted accordingly, and 2) the generation of each quotient digit is determined by only one digit in the partial remainder together with the transfer digit (or carry/borrow) emanating from it. A working example of radix 16 is given.**

Index Terms—**Deterministic generation of quotient, division, redundant number system, nonrestoring division, signed-digit arithmetic, signed-digit number system, Svoboda's division method.**

1. Introduction

In previous papers, A. Avizienis used a division algorithm based on Robertson's method [1] for signed-digit (S-D) arithmetic [2]–[4]. This note proposes another algorithm which is primarily derived from Svoboda's method [6]. S-D number representations are redundant positional representations. Some important and relevant properties of S-D representations are briefly described in the following.

In an S-D representation of radix r, each digit can assume values from a sequence of $(2a+1)$ integers:

$$\begin{aligned} &\{\bar{a}, \cdots, \bar{1}, 0, 1, \cdots, a\} \\ &\tfrac{1}{2}(r_o+1) \le a \le (r_o-1) \qquad \text{for odd radices } r_o \ge 3 \\ &(\tfrac{1}{2}r_e+1) \le a \le (r_e-1) \qquad \text{for even radices } r_e \ge 4, \end{aligned} \tag{1}$$

where the overbars are used to designate negative digit values. S-D numbers having minimal or maximal redundancy refer to the cases in which a in (1) assumes the legitimate minimal or maximal values, respectively.

The algebraic value Z of an S-D number z, $(z_{-n}, z_{-n+1}, \cdots, z_{-1}, z_0, z_1, \cdots, z_m)$, is given by

$$Z = \sum_{i=-n}^{m} z_i r^{-i} \tag{2}$$

and $Z=0$ if and only if $z_i=0$ for all i. The sign of Z is given by the sign of the most significant (left-most) nonzero digit in z. To form the representation of the additive inverse $-Z$, the sign of every nonzero digit z_i is changed individually.

The addition and subtraction of two S-D operands x and y satisfy

$$s_i = f(x_i, y_i, x_{i+1}, y_{i+1})$$

for all i, where

$$s = x + y \quad \text{or} \quad s = x - y.$$

This in turn implies that there is no carry-propagation chain in S-D addition or subtraction. Hence, the time of addition or subtraction of S-D numbers is independent of the length of the operands. The digit addition algorithm is performed in two steps. First, an interim sum w_i and a transfer digit t_{i-1} are computed by

$$w_i = x_i + y_i - rt_{i-1} \tag{3}$$

where

$$\begin{aligned} t_{i-1} &= 0 \quad \text{if } |x_i + y_i| < a \\ &= 1 \quad \text{if } x_i + y_i \ge a \\ &= \bar{1} \quad \text{if } x_i + y_i \le \bar{a}. \end{aligned}$$

Second, the sum digit s_i is obtained by

$$s_i = w_i + t_i. \tag{4}$$

Previous design studies have shown that maximally redundant S-D systems contain "pseudonormal" forms which demand special handling and increase the complexity of floating-point algorithms [5] and that maximally redundant forms have a wider range of multiplier digit values which needs a more complicated recoding of the multiplier for efficient multiplication; hence, minimally redundant S-D systems are used throughout the rest of this paper.

2. Svoboda's Division Method

Svoboda's division method consists of two steps. In Step 1, the input divisor, assumed normalized, is adjusted to be of the form $(1+e)$ where e is a positive fractional quantity, while the dividend is adjusted accordingly and normalized if necessary. Step 2 is described in Table I.

In Table I, the test of the equality of $x_{j+1}{}^{j+1} = x_{j+1}{}^{j}$ is to check if $x_{j+1}{}^{j}$, the prospective quotient digit, is correct or has caused overshoot. If this test turns out to be negative, the deviation caused by the overshoot must be compensated for in the immediately following step by an addition/subtraction of $r^{-(j+1)}e$ if the last step was a subtraction/addition. The requirement that this compensation be possible in one step puts a constraint on the range of e.

At the end of n iterations, the left n digits of x^n are those which have been used as if they were quotient digits, denoted as q, and the rest of x^n, denoted as d, may be considered the remainder. The arithmetic relation which characterizes the above algorithm is

$$x^n = x^0 - qe = q + d \tag{5}$$

or equivalently

$$x^0 = q(1+e) + d = qy + d. \tag{6}$$

Thus the above computation is verified to be a division.

An example of radix 10 where e is found to be less than 0.1 is given in Table II. It should be pointed out that the compensation is done, if needed, in a "restoring" manner. In his paper, Svoboda, employing complement forms for subtraction and detecting the carry (borrow) which changes the value of $x_{j+1}{}^{j}$, performs the division in a "nonrestoring" manner. Thus, the total number of additions in Step 2 is minimized to n, the length of the single-length operand. The "restoring" manner description of this method is used here solely for the sake of easier understanding.

3. A Modified Svoboda's Method for the S-D Arithmetic

3.1. *The Range of e*

The properties that the S-D numbers can be redundantly represented and each digit carries its own sign make Svoboda's method rather attractive to the S-D arithmetic. The range of e is determined in the following. As stated previously, the allowed digits of the minimally redundant S-D representations are

$$\begin{aligned} &\overline{(\tfrac{1}{2}r_e+1)}, \overline{(\tfrac{1}{2}r_e)}, \cdots, \bar{1}, 0, 1, \cdots, (\tfrac{1}{2}r_e), (\tfrac{1}{2}r_e+1) \\ &\qquad\qquad \text{for even radices } r_e \ge 4, \\ &\overline{(\tfrac{1}{2}(r_o+1))}, \overline{(\tfrac{1}{2}(r_o+1)-1)}, \cdots, \bar{1}, 0, 1, \cdots, \\ &\qquad (\tfrac{1}{2}(r_o+1)-1), (\tfrac{1}{2}(r_o+1)) \qquad \text{for odd radices } r_o \ge 3. \end{aligned} \tag{7}$$

3.1.1. *Upper Bounds:* The most misleading case which would result in a wrong estimate of the prospective quotient digit arises when the partial remainder assumes the format where for even radices the trial digit is $(\frac{1}{2}r_e+1)$ and all the digits to its right are $\overline{(\frac{1}{2}r_e+1)}$, or the inverse of this format. In order to correct in one step the deviation caused by such a wrong estimate, the following relation must be satisfied (assuming e is a positive fractional quantity):

Manuscript received April 3, 1967; revised May 24, 1968. The work reported here was supported by the Office of Naval Research and the Atomic Energy Commission under Contracts Nonr 233(52) and AT(11-1) Gen 10, Project 14, and by the Advanced Research Projects Agency under Contract SD 184.

The author was with the Department of Engineering, University of California, Los Angeles, Calif. He is now with IBM Research Lab., San Jose, Calif. 95114

TABLE I
SVOBODA'S DIVISION METHOD (STEP 2)

1) $j=0$.

2) $x^{j+1}=x^j-r^{-(j+1)}\cdot x_{j+1}{}^j\cdot e$.

3) If $x_{j+1}{}^{j+1}=x_{j+1}{}^j$ go to 5); otherwise go to 4).

4) $x^{j+1}=\mathrm{sgn}(x_{j+1}{}^j)\cdot r^{-(j+1)}\cdot e+x^{j+1}$.

5) If $j=n-1$ go to 7); otherwise go to 6).

6) $j=j+1$; go to 2).

7) Stop.

r: the radix; x^j: the jth partial remainder; $\mathrm{sgn}(x_j)=x_j/|x_j|$

TABLE II
AN EXAMPLE OF SVOBODA'S DIVISION METHOD (STEP 2)

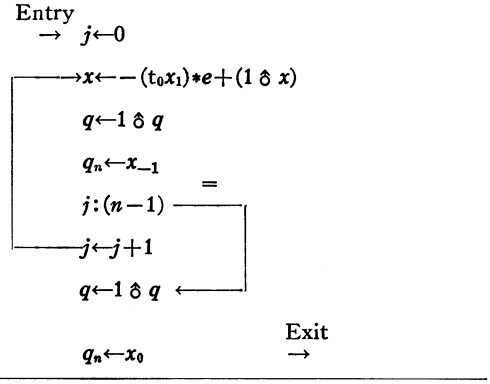
Radix 10
Dividend: 0.6500
Divisor: 1.0400

0.6 5 4 2 0 0 0 0	
−) 0.0 2 4 0 0 0 0 0	$6\cdot e\cdot 10^{-1}\cdots 6$
0.6 3 0 2 0 0 0 0	
−) 0.0 0 1 2 0 0 0 0	$3\cdot e\cdot 10^{-2}$
0.6 2*9 0 0 0 0 0	$3-1=2$
+) 0.0 0 0 4 0 0 0 0	$1\cdot e\cdot 10^{-2}$
0.6 2 9 4 0 0 0 0	
−) 0.0 0 0 3 6 0 0 0	$9\cdot e\cdot 10^{-3}\cdots 9$
0.6 2 9 0 4 0 0 0	
−) 0.0 0 0 0 0 0 0 0	$0\cdot e\cdot 10^{-4}\cdots 0$
0.6 2 9 0 4 0 0 0	

$0.6542/1.0400=0.6290+0.4000\cdot 10^{-4}$

* 3 is changed to 2.

TABLE III
THE ALGORITHM OF THE MODIFIED SVOBODA'S DIVISION METHOD (STEP 2) FOR THE S-D ARITHMETIC



Note: *: multiplication
k ô x: left shift by k digits
$j:n$: comparison of j and n.

TABLE IV
ONE IMPLEMENT OF STEP 1 OF THE MODIFIED SVOBODA'S DIVISION METHOD FOR THE S-D ARITHMETIC (RADIX 16, MINIMAL REDUNDANCY)

Divisor	Factor	Divisor	Factor
$0.1\bar{9}$–$0.1\bar{6}$	18.0	$1.\bar{6}\bar{9}$–$1.\bar{6}\bar{4}$	$2.\bar{5}$
$0.1\bar{5}$–0.11	$1\bar{1}.0$	$1.\bar{6}\bar{3}$–$1.\bar{6}1$	$2.\bar{6}$
0.12–$0.2\bar{4}$	9.0	$1.\bar{6}2$–$1.\bar{6}9$	1.9
$0.2\bar{3}$–$0.3\bar{5}$	6.0	$1.\bar{5}\bar{6}$–$1.\bar{5}0$	1.8
$0.3\bar{4}$–0.49	3.9	$1.\bar{5}1$–$1.\bar{4}\bar{9}$	1.7
$0.5\bar{6}$–$0.8\bar{4}$	2.2	$1.\bar{4}\bar{8}$–$1.\bar{4}0$	1.6
$0.8\bar{3}$–0.99	1.4	$1.\bar{4}1$–$1.\bar{3}\bar{6}$	1.5
		$1.\bar{3}\bar{5}$–$1.\bar{3}3$	1.4
		$1.\bar{3}4$–$1.\bar{2}\bar{1}$	1.3
		$1.\bar{2}0$–$1.\bar{1}\bar{5}$	1.2
		$1.\bar{1}\bar{4}$–$1.\bar{1}\bar{6}$	1.1
		$1.\bar{1}7$–1.09	1.0

$$\left| -(r^{-1}(\tfrac{1}{2}r+1)+r^{-2}(\tfrac{1}{2}r+1)+\cdots) - (\tfrac{1}{2}r+1)e \right| \le (1+e). \quad (8)$$

Thus, the upper bound of e is

$$e_{eu}=(r_e-4)/r_e(r_e-1) \quad \text{for even radices.} \quad (9)$$

Similarly, for odd radices

$$\left| -(r^{-1\frac{1}{2}}(r+1)+r^{-2\frac{1}{2}}(r+1)+\cdots) - \tfrac{1}{2}(r+1)e \right| \le (1+e). \quad (10)$$

Thus, the upper bound of e is

$$e_{ou}=(r_o-3)/(r_o-1)^2 \quad \text{for odd radices.} \quad (11)$$

3.1.2. *Lower Bounds:* Let h be a positive fractional quantity and e be $-h$. The most misleading situation in this case is that for even radices the trial quotient digit and all the digits to its right are $(\frac{1}{2}r_e+1)$, or the inverse of this format. Hence, the following relation must be satisfied:

$$\left| (r^{-1}(\tfrac{1}{2}r+1)+r^{-2}(\tfrac{1}{2}r+1)+\cdots) - (\tfrac{1}{2}r+1)(-h) \right| \le (1-h). \quad (12)$$

Thus, the lower bound of e is

$$e_{el}=-h_{e,\max}=-(r_e-4)/(r_e-1)(r_e+4) \quad \text{for even radices.} \quad (13)$$

Similarly, for odd radices

$$\left| (r^{-1\frac{1}{2}}(r+1)+r^{-2\frac{1}{2}}(r+1)+\cdots) - \tfrac{1}{2}(r+1)(-h) \right| \le (1-h). \quad (14)$$

Thus, the lower bound of e is

$$e_{ol}=-h_{o,\max}=-(r_o-3)/(r_o-1)(r_0+3) \quad \text{for odd radices.} \quad (15)$$

3.2. *The Algorithm*

Assume, without loss of generality, the input divisor is positive and normalized.

TABLE V
AN EXAMPLE ILLUSTRATING THE ALGORITHM IN TABLE III

Assume dividend $=0.9\bar{9}2\bar{6}$, $1+e=1.0900$, then

$q=0.8283$ and $d=0.6599\cdot 16^{-4}$

$q_0 \cdot q_1q_2q_3q_4$	$x_{-1}x_0 \cdot x_1x_2x_3x_4$	t_0	j	Next Action
0 0000	00 9 $\bar{9}$ 2 $\bar{6}$	0	0	form $-9*e$
	00 $\bar{5}$ $\bar{1}$ 00			$x \leftarrow 1 \delta x$
	09 $\bar{9}$ 2 $\bar{6}$ 0			add
	08 2 1 $\bar{6}$ 0	$\bar{1}$		$q \leftarrow 1 \delta q$
0 0000				$q_4 \leftarrow x_{-1}$
0 0000			1	form $-(\bar{1}2)*e$
	00 8 $\bar{2}$ 00			$x \leftarrow 1 \delta x$
	82 1 $\bar{6}$ 00			add
	82 9 $\bar{8}$ 00	0		$q \leftarrow 1 \delta q$
0 0000				$q_4 \leftarrow x_{-1}$
0 0008			2	form $-9*e$
	00 $\bar{5}$ $\bar{1}$ 00			$x \leftarrow 1 \delta x$
	29 $\bar{8}$ 000			add
	28 3 $\bar{1}$ 00	$\bar{1}$		$q \leftarrow 1 \delta q$
0 0080				$q_4 \leftarrow x_{-1}$
0 0082			3	form $-(\bar{1}3)*e$
	00 7 5 00			$x \leftarrow 1 \delta x$
	83 $\bar{1}$ 000			add
	83 6 5 00			$q \leftarrow 1 \delta q$
0 0820				$q_4 \leftarrow x_{-1}$
0 0828				$q \leftarrow 1 \delta q$
0 8280				$q_4 \leftarrow x_0$
0 8283				

Step 1: Multiply the divisor by a series of appropriate constant factors such that it becomes $(1+e)$ and e is bounded by (9) and (13) or (11) and (15) while the dividend is adjusted accordingly and normalized again. The actual process certainly depends on the individual implementation environment. An example is shown in the next section.

Step 2: For deeper understanding of the digit-wise operations involved this step is described by a program written in Iverson's language [7], [8] shown in Table III in which all words (operands) are treated as vectors. At the end of Step 1, the divisor is of the form

$$1 \cdot e_1e_2 \cdots e_n,$$

and the dividend is

$$x = x_{-1}x_0 \cdot x_1x_2 \cdots x_m$$

with $x_{-1}=x_0=0$; m may be n or $2n$, where n is the length of the single-length operand. q is the quotient and t_0 is the transfer digit or carry/borrow into the 0th position and can be $\bar{1}$,0,1.

3.3. *A Working Example*

An example of radix 16 and minimum-redundancy is given here to illustrate this algorithm.

1) The bounds adopted here for e are 0.0375 ($0.0999\cdots$ in radix 16) and -0.0375 ($0.0\overline{999}\cdots$ in radix 16). The bounds are symmetrical and are within the theoretical bounds derived above. The divisor is brought into this required range in two substeps. In each substep the first two fractional digits are examined and an appropriate factor is multiplied to the divisor. This is shown in Table IV. The bounds and substeps used here are by no means the best or optimum ones. They only serve the purpose of illustrating how Step 1 of this algorithm can be implemented.

2) Table V shows how Step 2 can be mechanized according to the rules in Table III.

4. CONCLUSION

A division algorithm, particularly suitable for floating-point arithmetic, has been described for S-D number systems. The speed of Step 1 of this algorithm is independent of the length of operands; however, the speed of Step 2 is nearly proportional to the length of operands. The introduction of Step 1 makes the fast generation of quotient in Step 2 possible. Generally speaking, the advantages of this method will be more appreciated when longer operands are dealt with.

In comparison, Robertson's method does not have the preprocessing of the divisor as does Step 1 of this method but it requires comparisons of the multiples of the divisor and the partial remainder for the generation of each individual quotient-digit—a process that is certainly more expensive than Step 2 of the modified Svoboda's method. Therefore, the modified Svoboda's method will be preferred when the economy gained in Step 2, with respect to Robertson's method, can more than compensate for the complexity introduced by Step 1.

ACKNOWLEDGMENT

The author wishes to express his sincere thanks to Prof. Svoboda and Prof. Avizienis for their enlightening suggestions and warm encouragement.

REFERENCES

[1] J. E. Robertson, "A new class of digital division method," *IRE Trans. Electronic Computers*, vol. EC-7, pp. 218–222, September 1958.
[2] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. Electronic Computers*, vol. EC-10, pp. 389–400, September 1961.
[3] ——, "On a flexible implementation of digital computer arithmetic," in *Information Processing*, C. M. Popplewell, Ed. Amsterdam, Netherlands: North-Holland Publishing Co., 1963, pp. 664–670.
[4] ——, "Arithmetic microsystems for the synthesis of function generators," *Proc. IEEE* (Special Issue on Computers), pp. 1910–1919, December 1966.
[5] A. Avizienis and D. Kimble, "A general building block for digital arithmetic," *Proc. Nat'l Symp. on the Impact of Batch Fabrication on Future Computers* (Los Angeles, Calif., April 6–8, 1965), pp. 173–180.
[6] A. Svoboda, "An algorithm for division," *Information Processing Machines* (Prague, Czechoslovakia), no. 9, 1963.
[7] K. Iverson, *A Programming Language*. New York: Wiley, 1962.
[8] ——, "Programming notation for systems design," *IBM Sys. J.*, vol. 2, pp. 117–128, June 1963.

An Analysis of High-Speed, Linear-Passive Binary, READ-Only Stores

RODGER L. GAMBLIN

***Abstract*—A general analysis of conventional high-speed, linear-passive, binary, READ-only stores is performed by the application of the scattering-matrix formalism. It is found that for a broad class of cases, which include most READ-only stores that have been discussed in the literature, the maximum transmission of power from a**

Manuscript received November 22, 1967; revised May 27, 1968.
The author is with IBM Corporation, Systems Development Division, Endicott, N. Y.