output of one adder is not fed into the carry input of the following adder so that most of the adders have to be implemented using a single adder or two-bit adders rather than the more economical 4-bit adders. Dadda's scheme is more systematic and more 4-bit adders are used in full capacity. As shown in Fig. 5, the structure of the carry–save multiplier is very systematic and well suited for using 4-bit adders.

We conclude that if the number of bits in the multiplicand or multiplier is small, the carry–save scheme using a ripple adder is favored over the other two schemes. It is more economical, less complicated, and the difference in speed is not significant. However, as n increases, the Dadda's multiplier gets increasingly faster than the carry–save multiplier. It uses fewer components than Wallace's multiplier and operates at the same speed; hence it is always better than Wallace's multiplier.

IV. The Purdue Fast Multiplier

In this section we describe the design, construction, and performance of two fast multipliers implemented at the PCM Telemetry Laboratory, Department of Electrical Engineering, Purdue University. These identical multipliers were designed as parts of two digital filters that, in turn, are part of an experimental PCM system [5]. The experimental PCM system is an implementation of the optimum PCM system suggested by Wintz and Kurtenbach [6].

The multiplier is designed to multiply a 12-bit multiplicand by a 10-bit multiplier. It uses the carry–save scheme of reducing the summands and a ripple adder to add the final two numbers. The unit was constructed from Texas Instruments 74 series integrated circuits. The wiring was done on perforated double-size boards manufactured by Digital Equipment Corporation and dual-in-line integrated circuit sockets are used for ease in replacement of modules (see Fig. 6). The whole multiplier occupies three double-size boards and because of limitations in the number of input and output connections it is convenient that parts of the elements of the summand matrix are generated along with the circuitry producing the final product on each board.

The worst case multiplication time is about 520 ns; this is in agreement with the 495 ns predicted. The multiplication time could be decreased to 460 ns by replacing the 74 series logic gates with 74 H series since the propagation delay of the 74 H series is only half that of the 74 series. This would increase the cost of the multiplier from $456 to $550. The multiplication time could be decreased to 410 ns by using an 11-bit carry–look-ahead adder constructed from 74 H series for adding the final two numbers. This will boost the cost to $710. Using Dadda's scheme this unit would have cost about $770 and the multiplication time would have been 360 ns. Replacing the 74 series with the 74 H series would also reduce the multiplication time of Dadda's system to 270 ns while increasing its cost to $1010.

Acknowledgment

The authors are grateful to the referees for their valuable suggestions, and in particular, to one of them for his very constructive criticisms.

References

[1] L. Dadda, "Some schemes for parallel multipliers," *Alta Frequenza*, vol. 34, pp. 349–356, March 1965.
[2] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electronic Computers*, vol. EC-13, pp. 14–17, February 1964.
[3] E. L. Braun, *Digital Computer Design*. New York: Academic Press, 1963.
[4] F. Mowle, lecture notes on a course in digital computer design techniques, Purdue University, Lafayette, Ind., 1967.
[5] G. G. Apple and P. A. Wintz, "Experimental PCM system employing Karhunen–Loeve sampling," presented at the 1969 Internatl. Symp. on Information Theory, Ellenville, N. Y., January 1969.
[6] P. A. Wintz and A. J. Kurtenbach, "Waveform error control in PCM telemetry," *IEEE Trans. Information Theory*, vol. IT-14, pp. 650–661, September 1968.

On Range-Transformation Techniques for Division

E. V. KRISHNAMURTHY

***Abstract*—This note points out the close relationship between some of the recently described division techniques, in which the divisor is transformed to a range close to unity. A brief theoretical analysis is presented which examines the choice of quotient digit when this type of division technique is used for conventional and signed-digit number systems.**

***Index Terms*—Conventional and signed-digit number systems, deterministic generation of quotient digit, divide and correct methods, Harvard iterative technique, nonrestoring division, precision of multiplication, range transformation of divisor, Svoboda's method, Tung's algorithm.**

I. Introduction

Recently a number of techniques for division have been described which consist in initially transforming the divisor to a suitable range by premultiplication, so that the choice of the quotient digit is deterministic without any need for a trial and error process [1], [6]. Although all these techniques are closely related, since each one of them has been discovered independently and reported in different journals at about the same time, it is natural that little or no attention has been paid in bringing out the close relationship that exists between them. It is the object of this note to bring out this relationship and place all these techniques on a common basis with the hope that it would be useful for workers in this field.

II. General Development

Let us denote the dividend A and divisor B in floating-point form with integral mantissa in radix β thus:

$$A = \beta^{e_a} \cdot a = \beta^{e_a} \cdot \sum_{j=0}^{n} a_j \beta^j \tag{1a}$$

Manuscript received February 20, 1969; revised June 27, 1969.
The author is with the Weizmann Institute of Science, Department of Applied Mathematics, Rehovot, Israel, on leave of absence from the Indian Statistical Institute, Calcutta, India.

$$B = \beta^{e_b} \cdot b = \beta^{e_b} \cdot \sum_{j=0}^{d} b_j \beta^j. \tag{1b}$$

Here e_a, e_b are the exponents and a, b are the mantissa of A and B, respectively.

Let us for convenience assume that we are interested in a quotient Q whose mantissa q has the same precision as a. Thus

$$Q = \beta^{e_q} \cdot q = \beta^{e_q} \sum_{j=0}^{n} q_j \beta^j. \tag{1c}$$

(Note that the exponent e_q is to be suitably determined [1].)

It is now interesting to study under what conditions the quotient digit q_j can be chosen as the leading (or most significant) digit of every partial remainder (for the first time, the dividend) plus a suitable constant, say, δ positive or negative. In Nandi and Krishnamurthy [1], two schemes which arose as a special case of the general divide and correct methods discussed in [2] and [3] are described.

In the first scheme q_j is chosen as the leading digit of the partial remainder, if the divisor starts with the digit 1 and is followed by a 0 in the next lower order position, while in the second scheme q_j is chosen as the leading digit of the partial remainder plus unity, if the divisor starts with the digit $(\beta-1)$. It is to be noted that the first one of these schemes is the same as Svoboda's scheme [4].

For the purpose of our study, let us denote by R_j the partial remainder at each step with $R_j(j=n)=a$, the mantissa of the dividend.

The usual recursion is then (see [1])

$$R_{j-1} = R_j - q_j b \beta^{j-d-1}. \tag{2}$$

Assuming that we carry out a nonrestoring division operation, if a_j represents the leading digit of R_j, and we need $q_j = a_j + \delta$ to be the true nonrestoring quotient, then R_{j-1} has to satisfy

$$|R_{j-1}| = |R_j - (a_j + \delta) b \beta^{j-d-1}| < b\beta^{j-d-1} \tag{3}$$

from which one readily obtains that b should be in the range

$$\frac{R_j}{a_j + \delta + 1} < b\beta^{j-d-1} < \frac{R_j}{a_j + \delta - 1} \cdot \tag{4}$$

Since we are interested in a divisor b which would satisfy this condition for all R_j, we must consider the greatest of this lower bound and the least of this upper bound to determine the range of b. We will now consider these for conventional and signed-digit number systems and the possible choices of δ.

III. Conventional Number Systems

Case 1: $\delta = 0$.

We need

$$\left(\frac{R_j}{a_j + 1}\right)_{\max} < b\beta^{j-d-1} < \left(\frac{R_j}{a_j - 1}\right)_{\min}. \tag{5}$$

Note

$$\left(\frac{R_j}{a_j + 1}\right)_{\max} = \left(\frac{a_j\beta^j + a_{j-1}\beta^{j-1} + \cdots}{a_j + 1}\right)_{\max} \mp \beta^j \tag{6}$$

and

$$\left(\frac{R_j}{a_j - 1}\right)_{\min} = \left(\frac{a_j\beta^j}{a_j - 1}\right)_{\min} = \frac{(\beta - 1)}{(\beta - 2)} \beta^j. \tag{7}$$

(Here $\mp$ denotes approximately close to.) Note $R_j = 0$ implies $a_j = 0$, hence $q_j = 0$; we must omit this case in finding the bound in (7). Also note that as in [1]–[3], we do not insist in (3) that the coefficient or digit in β^jth position of R_{j-1} becomes zero; in such a case (7) is more stringent, i.e.,

$$\left(\frac{R_j}{a_j - 1}\right)_{\min} \frac{(\beta+1)}{\beta} \beta^j.$$

This scheme is useful if one wants q_j to be stored in the vacant storage (β^jth) of R_{j-1}. (See [10], [11].) Now (5) can be rewritten as

$$1 < b\beta^{-(d+1)} < \frac{\beta - 1}{\beta - 2} \cdot \tag{8}$$

This means the mantissa b must be transformed by suitable multiplication to range (8) with a suitable change in e_b, q, and e_q (or a and e_a).

Case 2: $\delta = 1$.

We need

$$\left(\frac{R_j}{a_j + 2}\right)_{\max} < b\beta^{j-d-1} < \left(\frac{R_j}{a_j}\right)_{\min} \cdot \tag{9}$$

Note

$$\left(\frac{R_j}{a_j + 2}\right)_{\max} \mp \frac{\beta}{\beta + 1} \beta^j \tag{10}$$

and

$$\left(\frac{R_j}{a_j}\right)_{\min} = \beta^j. \tag{11}$$

(When $R_j = 0$, R_j/a_j is undetermined; we exclude the case $R_j = 0$ to obtain (11). For this case we take $q_j = 0$. See [1].) Thus (9) can be rewritten as

$$\frac{\beta}{\beta + 1} < b\beta^{-(d+1)} < 1. \tag{12}$$

(Here $(d+1)$ is the number of digits in b, as seen from (1b).)

Note, however, the lower bound in (12) can be improved to $(\beta-1)/\beta$ by setting initially a_n, the leading digit of the dividend, as zero and taking care of the overflow of the quotient [1]. Hence we obtain,

$$\frac{\beta-1}{\beta} \leq b\beta^{-(d+1)} < 1. \tag{12a}$$

This means the mantissa b must be transformed by suitable multiplication to range (8) with a suitable change in e_b, q, and e_q (or a and e_a).

Case 3: $\delta \geq +2$ and $\delta \leq -1$.

It is easy to see that for these cases the inequality

$$\left(\frac{R_j}{a_j+\delta+1}\right)_{\max} < \left(\frac{R_j}{a_j+\delta-1}\right)_{\min}$$

cannot be satisfied for all R_j.

Thus for conventional number systems there are two possible schemes for deterministic generation of the quotient digit according as b satisfies (8) or (12). These have been reported in [1].

IV. Signed-Digit Number Systems

In this case, let us assume that γ is the highest digit magnitude in a and b [5], [6].

Case 1: $\delta = 0$.

We need

$$\left(\frac{R_j}{a_j+1}\right)_{\max} < b\beta^{j-d-1} < \left(\frac{R_j}{a_j-1}\right)_{\min}. \tag{13}$$

Now

$$\begin{aligned}\left(\frac{R_j}{a_j+1}\right)_{\max} &\mp \frac{\gamma(1+1/(\beta-1))}{\gamma+1}\beta^j \\ &= \frac{\gamma}{\beta-1}\,\frac{\beta}{\gamma+1}\beta^j\end{aligned} \tag{14}$$

and

$$\begin{aligned}\left(\frac{R_j}{a_j-1}\right)_{\min} &\mp \frac{\gamma(1-1/(\beta-1))}{\gamma-1}\beta^j \\ &= \frac{\gamma}{\beta-1}\,\frac{\beta-2}{\gamma-1}\beta^j.\end{aligned} \tag{15}$$

Thus (13) can be rewritten as

$$\frac{\gamma}{\beta-1}\,\frac{\beta}{\gamma+1} < b\beta^{-(d+1)} < \frac{\gamma}{\beta-1}\,\frac{\beta-2}{\gamma-1}. \tag{16}$$

This means b has to be transformed by suitable multiplication to range (16) with suitable changes in e_b, q, and e_q (or a and e_a).

Case 2: $\delta = 1$.

We need

$$\left(\frac{R_j}{a_j+2}\right)_{\max} < b\beta^{j-d-1} < \left(\frac{R_j}{a_j}\right)_{\min}. \tag{17}$$

Note

$$\begin{aligned}\left(\frac{R_j}{a_j+2}\right)_{\max} &= \frac{\gamma}{\gamma+2}\left(1+\frac{1}{(\beta-1)}\right)\beta^j \\ &= \frac{\gamma\beta}{(\beta-1)(\gamma+2)}\beta^j.\end{aligned} \tag{18}$$

However, $(R_j/a_j) < 1\cdot\beta^j$ unlike the conventional number system where this quantity is always greater than $1\cdot\beta^j$ (see (11)). In fact

$$\left(\frac{R_j}{a_j}\right)_{\min} = \left(1-\frac{\gamma}{(\beta-1)}\right)\beta^j = \frac{\beta-1-\gamma}{\beta-1}\cdot\beta^j. \tag{19}$$

Thus we obtain from (17), using (18) and (19),

$$\frac{\gamma\beta}{(\beta-1)(\gamma+2)} < \frac{\beta-1-\gamma}{\beta-1}$$

or

$$(\gamma+1)(\gamma+2) < 2\beta \tag{20}$$

which in general cannot be satisfied for any positive integral radix, when γ_{min} has to satisfy the conditions

$$(\beta-1) \geq \gamma \geq \tfrac{1}{2}(\beta+1) \qquad \text{for } \beta \text{ odd}$$

$$(\beta-1) \geq \gamma \geq \tfrac{1}{2}\beta+1 \qquad \text{for } \beta \text{ even.}$$

Case 3: $\delta \geq +2$ and $\delta \leq -1$.

It is easy to see that for these cases the inequality

$$\left(\frac{R_j}{a_j+\delta+1}\right)_{\max} < \left(\frac{R_j}{a_j+\delta-1}\right)_{\min}$$

cannot be satisfied for all R_j.

Thus for signed-digit number systems the choice $q_j = a_j$ is possible provided the divisor satisfies (16). (For alternative choices, additional information, namely a knowledge of the sign or magnitude of a_{j-1}, is needed; see [10], [11].)

V. Remarks

It is worthwhile noting that a conventional number system offers a wider range of divisor for deterministic generation of the quotient digit; this range is (combining the contiguous intervals (12) and (8))

$$\frac{\beta-1}{\beta} \leq b\beta^{-(d+1)} < \frac{\beta-1}{\beta-2}. \tag{21}$$

For example, for $\beta = 16$, (21) can be written as

$$\frac{15}{16} \leq b\beta^{-(d+1)} < \frac{15}{14}. \tag{22}$$

While from (16) for $\beta=16$, $\gamma=9$ we obtain

$$\frac{24}{25} < b\beta^{-(d+1)} < \frac{21}{20}. \qquad (23)$$

(Compare (23) with Tung's [6] result.)

It is interesting to note that the range transformation schemes are analogous to the Harvard iterative scheme [7] (see also Wallace [8]) in which the divisor is successively multiplied so as to be transformed to a number arbitrarily close to unity, as desired. The only difference, as explained in [1], is that this initial range transformation together with a nonrestoring division scheme is more economical, since the numerator need not be multiplied more than once. However, the choice of a suitable multiplier initially poses a problem and this has to be considered to evaluate the relative merits. Some studies have been made by the author regarding optimal precision requirement and choice of the multipliers for the direct as well as iterative schemes. (See [9], [12].)

In this context, it is worthwhile noting that the interval width or range of b in (16) contracts as γ increases, with a zero width for $\gamma=\beta-1$. As observed elsewhere [9], the range transformation of numbers involves greater computational labor if the range width decreases; thus Tung's algorithm becomes unsuitable as redundancy increases. Also, in Tung's algorithm the quotient digit obtained can have a magnitude $(\beta-1)$, larger than γ, thereby requiring facility for storing a two-digit quotient and a facility for two-digit multiplication for computing the partial remainder. A different analysis made by the author to remove these difficulties has given rise to more efficient algorithms; these are available in [10] and [11], and will appear elsewhere.

References

[1] S. K. Nandi and E. V. Krishnamurthy, "A simple technique for digital division," *Commun. ACM*, vol. 10, pp. 299–301, May 1967.

[2] E. V. Krishnamurthy, "On a divide-and-correct method for variable precision division," *Commun. ACM*, vol. 8, pp. 179–181, March 1965.

[3] E. V. Krishnamurthy and S. K. Nandi, "On the normalization requirement of divisor in divide and correct methods," *Commun. ACM*, vol. 10, pp. 809–813, December 1967.

[4] A. Svoboda, "An algorithm for division," *Information Processing Machines* (Prague, Czechoslovakia), no. 9, 1963.

[5] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. Electronic Computers*, vol. EC-10, pp. 389–400, September 1961.

[6] C. Tung, "A division algorithm for signed-digit arithmetic," *IEEE Trans. Computers* (Short Notes), vol. C-17, pp. 887–889, September 1968.

[7] R. K. Richards, *Arithmetic Operations in Digital Computers*. London: Van Nostrand, 1955.

[8] C. S. Wallace, "A suggestion for fast multiplier," *IEEE Trans. Electronic Computers*, vol. EC-13, pp. 14–17, February 1964.

[9] E. V. Krishnamurthy and B. P. Sarkar, "Algorithm for multiple-precision range transformation," *Sankhyā (B)*, vol. 30, parts 1 and 3, pp. 33–46, June 1968.

[10] E. V. Krishnamurthy, "A more efficient range-transformation algorithm for signed-digit division," *Internatl. J. of Control*, to be published.

[11] ——, "Carry-borrow free sequential quotient generation with segmented signed-digit operands," *Internatl. J. of Control*, to be published.

[12] ——, "On optimal iterative schemes for high-speed division," *IEEE Trans. Computers*, vol. C-19, March 1970.

Pictorial Output with a Line Printer

I. D. G. MACLEOD

***Abstract*—An improved method for the production of pictorial output on a line printer is described. A reasonable black-white contrast ratio is obtained by overprinting up to eight characters, and pseudorandom noise is used to smooth out discontinuities in the range of print densities.**

***Index Terms*—Linear interpolation, line printers, overprinting, pictorial output, picture coding, picture output.**

The possible character positions on line printer output may be regarded as cells in a two-dimensional array. By choosing the character (or combination of overprinted characters) printed in each cell on the basis of average print density, a pictorial representation of any desired two-dimensional data may be generated. The maximum number of printed characters in a line is a restriction to the horizontal size of the picture unless it is assembled from several strips. Pictures produced in this manner will be inferior to those produced by special-purpose hardware, but the convenience and ready availability of a line printer will in many cases outweigh any loss of quality.

Perry and Mendelsohn [1] describe such a method of picture generation with a line printer. They use pairs of adjacent character positions in each printed line as basic density cells, and overprint a maximum of two characters. We have found that a greater degree of overprinting yields a worthwhile increase in maximum density (and thus contrast) without an undue penalty in printout time on the IBM 1403 line printer (1100 lpm max.) attached to the Australian National University's IBM 360/50 computer. Using a maximum of eight overprinted characters, the output shown in Fig. 2(b) took approximately one minute to generate on this printer. The use of individual (rather than pairs of) character positions as basic density cells results in finer resolution and allows more information to be represented on each line. With the 1403 printer set to ten characters per inch horizontally and eight lines per inch vertically, noticeable distortion results from the difference in horizontal and vertical scales. This distortion is reduced by linearly interpolating output lines between rows of the data array, printing only eight lines for every ten rows in the data array.

A difficulty in determining the density codes (i.e., overprinted character combinations) to be employed is that it is not possible, given a conventional character set, to choose combinations such that there is a smooth transition from white to black, without perceptible discontinuities. The sudden transition from white (blank) to the next lightest combination (comma) is noticeable in the examples of pictorial output given in [1]. The problem of representing numerical data with density

Manuscript received April 18, 1969; revised August 27, 1969.
The author is with the Department of Engineering Physics, Australian National University, A.C.T. 2600, Australia.