Uniform Shift Multiplication Algorithms Without Overflow

PAOLO CORSINI AND GRAZIANO FROSINI

***Abstract*—In this correspondence the problem of performing the multiplication by recoding the multiplier is considered. A special recoding for fractional numbers in two's complement form is presented, that generates a class of uniform shift multiplication algorithms having the property that every partial product is always in the open interval $(-1,1)$. Both the scan of the multiplier from the least to the most significant bit and the scan in the opposite direction are considered.**

***Index Terms*—Higher radix multiplication, modified Booth's algorithms, multiplication algorithms, two's complement arithmetic, uniform shift methods.**

I. Introduction

In digital systems that operate with real-time data arriving at uniformly spaced time instants, it is useful to implement algorithms for the arithmetic operations that require a constant operation time, independent of the actual data values. Uniform shift algorithms for multiplication have been proposed [1]–[3] that do not require sign correction for numbers represented in two's complement form. An algorithm of this class examines at every step $Q + 1$ bits of an N bit multiplier, and adds a proper multiple of the multiplicand to the previous partial product, thus obtaining a new partial product. The final product is obtained after $M = N/Q$ steps.

In some applications, where fractional arithmetic is used, it is advantageous to have uniform shift algorithms in which at every step the partial product is always in the open interval $(-1,1)$, so that an overflow never occurs. This implies that the quantities to be added to the partial product must be fractions of the multiplicand or the multiplicand itself.

In this correspondence the problem of the multiplier recoding is considered [4]–[6], and a special recoding is given that allows us to represent the multiplier in radix 2^Q by means of M signed digits, where each digit is in the closed interval $(-1,1)$. Two multiplication algorithms are then derived (one corresponds to the scan of the multiplier from the least to the most significant bit, and the other to the scan in the opposite direction), having the property that every partial product is always in the open interval $(-1,1)$. Such algorithms, that represent a class of uniform shift multiplication algorithms, for $Q = 1$ coincide with the classical Booth algorithms [1], while for $Q = 2$ and for $Q = 3$ represent a modification of the "uniform shift of two" and "uniform shift of three" MacSorley algorithms [2].

II. Multiplier Recoding and Multiplication Algorithms

Let X and Y be two fractional numbers in two's complement form, and let us consider the product P, xy. If $X = Y = -1$ the product cannot be represented in fractional form and an overflow occurs. If either $X = -1$ or $Y = -1$ (but not both), then P is obtained simply by two's complementing Y or X. If $-1 < (X,Y) < 1$, then the product must be evaluated.

Referring to the last case, let $y_0,y_1,y_2,\cdots,y_{N-1}$ be the binary sequence representing the multiplier Y, y_0 being the sign bit and y_{N-1} the least significant bit. It results

$$Y = -y_0 + \sum_{n=1}^{N-1} y_n 2^{-n}. \tag{1}$$

In order to evaluate P, let us recode arithmetically the multiplier Y [4]–[6] so that it can be expressed as

$$Y = \sum_{m=0}^{M-1} z_m r^{-m},$$

where r is the new radix and z_m, $m = 0,1,\cdots,M-1$, are proper signed digits. The product $P = XY$ can be computed in M steps by evaluating M partial products, the Mth of which coincides with P. The partial products can be obtained by scanning the signed digits of the recoded multiplier either from the right to the left or in the opposite direction.

Right to Left Scan: Let us define as partial products the quantities

$$PR_i = X \sum_{m=M-i}^{M-1} z_m r^{M-i-m}, \qquad i = 1,2,\cdots,M.$$

Obviously $PR_M = P$. Moreover, the following iterative relation holds:

$$PR_i = r^{-1}PR_{i-1} + z_{M-i}X, \qquad i = 1,2,\cdots,M, \tag{2}$$

where $PR_0 = 0$. This relation represents an M-step multiplication algorithm in which the digits of the recoded multiplier are examined from the least to the most significant.

Left to Right Scan: Let us define as partial products the quantities:

$$PL_i = X \sum_{m=0}^{i-1} z_m r^{-m}, \qquad i = 1,2,\cdots,M.$$

Obviously $PL_M = P$. Moreover the following iterative relation holds:

$$\begin{aligned} PL_i &= PL_{i-1} + z_{i-1}X_{i-1}, \\ X_i &= r^{-1}X_{i-1}, \end{aligned} \qquad i = 1,2,\cdots,M, \tag{3}$$

where $PL_0 = 0$ and $X_0 = X$. Also this relation represents an M-step multiplication algorithm, and the digits of the recoded multiplier are examined from the most to the least significant.

III. A Special Recoding of the Multiplier

The problem of recoding the multiplier, in order to obtain multiplication algorithms having particular properties, has been extensively studied in the literature [4]–[6]. The goal of the proposed recodings has been chiefly to obtain a multiplier representation having a minimum number of nonzero digits, so that the time required to perform a specific multiplication is minimized.

In some applications it is advantageous to implement multiplication algorithms requiring a constant operation time, independent of the actual operand values. This happens when operations with real-time data arriving at uniformly spaced time instants are to be performed, and when a multiple data stream is processed under the control of a single instruction stream (array processors). In order to have a constant multiplication time, it is necessary that each multiplication step consists always of the same operations, i.e., of a shift operation and of an add operation, even if one of the addends is equal to 0.

Referring to the above applications, we propose a multiplier recoding having the property that every partial product is always in the open interval $(-1,1)$, so that an overflow never occurs. The signed digits by which the multiplier is recoded are not necessarily integers, and are constructed by scanning the multiplier bits $y_0,y_1,\cdots,y_{N-1}$ either from the right to the left or from the left to the right. Each one of the M signed digits is constructed by examining $Q + 1$ bits of the multiplier (where $MQ = N$) and two consecutive signed digits are functions of two sets of bits overlapped of one bit.

Manuscript received January 21, 1976; revised July 26, 1976. This work was financially supported by the Convention of Selenia S.p.A. and Consiglio Nazionale delle Ricerche, Pisa, Italy.

P. Corsini is with the Department of Electrical and Electronic Engineering, University of Pisa, Pisa, Italy.

G. Frosini is with the Istituto di Elaborazione della Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy.

0018-9340/78/0300-0256$00.75

Theorem 1: Let $y_0, y_1, \cdots, y_{N-1}$ be the binary sequence representing the multiplier Y and let us append to the right of the least significant bit y_{N-1} an additional bit $y_N = 0$. If $N = MQ$, M and Q being integers, then the multiplier can be recoded in radix $r = 2^Q$ and expressed as

$$Y = \sum_{m=0}^{M-1} z(Q)_m (2^Q)^{-m},$$

where

$$z(Q)_m = -y_{Qm} + \sum_{q=1}^{Q-1} y_{Qm+q} 2^{-q} + y_{Qm+Q} 2^{-Q+1}. \quad (4)$$

Proof: Relation (1) can be rewritten as

$$Y = -2y_0 + \sum_{n=0}^{N-1} y_n 2^{-n}.$$

Putting $n = Qm + q$, $q = 0,1,\cdots,Q-1$, $m = 0,1,\cdots,M-1$, we have

$$Y = -2y_0 + \sum_{q=0}^{Q-1}\sum_{m=0}^{M-1} y_{Qm+q} 2^{-Qm-q}$$
$$= -2y_0 + \sum_{m=0}^{M-1} y_{Qm} 2^{-Qm} + \sum_{q=1}^{Q-1}\sum_{m=0}^{M-1} y_{Qm+q} 2^{-Qm-q}.$$

By adding and subtracting the first summation we obtain

$$Y = -\sum_{m=0}^{M-1} y_{Qm} 2^{-Qm} + \sum_{m=0}^{M-1}\left(\sum_{q=1}^{Q-1} y_{Qm+q} 2^{-q}\right) 2^{-Qm} + 2\sum_{m=1}^{M-1} y_{Qm} 2^{-Qm}.$$

In the last summation the index m first can be extended until M, since in the theorem hypothesis $y_{MQ} = y_N = 0$, and then can be changed by putting $m = m - 1$, so that we have

$$Y = \sum_{m=0}^{M-1}\left(-y_{Qm} + \sum_{q=1}^{Q-1} y_{Qm+q} 2^{-q} + y_{Qm+Q} 2^{-Q+1}\right)(2^Q)^{-m}.$$

Q.E.D.

From (4) it follows that $-1 \le z(Q)_m \le 1$. As particular cases we have:

$$z(1)_m = -y_m + y_{m+1},$$
$$z(2)_m = -y_{2m} + \frac{1}{2} y_{2m+1} + \frac{1}{2} y_{2m+2},$$
$$z(3)_m = -y_{3m} + \frac{1}{2} y_{3m+1} + \frac{1}{4} y_{3m+2} + \frac{1}{4} y_{3m+3},$$

so that

$$z(1)_m \in \{-1, 0, 1\},$$
$$z(2)_m \in \left\{-1, -\frac{1}{2}, 0, \frac{1}{2}, 1\right\},$$
$$z(3)_m \in \left\{-1, -\frac{3}{4}, -\frac{1}{2}, -\frac{1}{4}, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right\}.$$

Theorem 2: If the multiplier is recoded according to Theorem 1, then every partial product PR_i (PL_i) satisfies to the relation $-1 < PR_i < 1$ ($-1 < PL_i < 1$).

Proof: The proof is made for every partial product PR_i. The proof for the partial products PL_i is similar and is omitted. In the theorem hypothesis we have

$$PR_i = X \sum_{m=M-i}^{M-1} z(Q)_m (2^Q)^{M-i-m}$$
$$= X \sum_{m=M-i}^{M-1}\left(-y_{Qm} + \sum_{q=1}^{Q-1} y_{Qm+q} 2^{-q} + y_{Qm+Q} 2^{-Q+1}\right)(2^Q)^{M-i-m}$$
$$= X \sum_{m=M-i}^{M-1}\left(-2y_{Qm} + \sum_{q=0}^{Q-1} y_{Qm+q} 2^{-q} + y_{Qm+Q} 2^{-Q+1}\right)(2^Q)^{M-i-m}$$
$$= X\left(-2\sum_{m=M-i}^{M-1} y_{Qm} 2^{-Qm} + \sum_{m=M-i}^{M-1}\sum_{q=0}^{Q-1} y_{Qm+q} 2^{-Qm-q} + 2\sum_{m=M-i}^{M-1} y_{Q(m+1)} 2^{-Q(m+1)}\right)(2^Q)^{M-i}.$$

By putting $n = Qm + q$ (so that $n = N - iQ, N - iQ + 1, \cdots, N - 1$) and by simplifying the elements of the first and of the last summation, we obtain

$$PR_i = X\left(-2y_{N-iQ} + \sum_{n=N-iQ}^{N-1} y_n 2^{N-iQ-n} + y_N 2^{-iQ+1}\right)$$
$$= X\left(-y_{N-iQ} + \sum_{n=N-iQ+1}^{N-1} y_n 2^{N-iQ-n}\right).$$

The quantity in parentheses is greater than or equal to -1 and less than 1. As $-1 < X < 1$, it follows that $-1 < PR_i < 1$.

Q.E.D.

IV. Concluding Remarks

Let us substitute the signed digits (4) and the radix 2^Q into relation (2). The following uniform shift of Q multiplication algorithm results to be defined:

$$PR_i = 2^{-Q} PR_{i-1} + z(Q)_{M-i} X, \qquad i = 1,2,\cdots,M, \quad (5)$$

where $PR_0 = 0$. The ith step of the algorithm is as follows:

1) the previous partial product is arithmetically right shifted Q places (initially the partial product is zero);

2) the multiplier bits are scanned from the right to the left, in order to construct $z(Q)_{M-i}$; and

3) the multiplicand X first is multiplied by $z(Q)_{M-i}$ and is then added to the quantity obtained from 1), so forming a new partial product.

Relation (5) for $Q = 1$ becomes

$$PR_i = \frac{1}{2} PR_{i-1} + (-y_{N-i} + y_{N-i+1})X, \qquad i = 1,2,\cdots,N$$

and represents the classical single shift Booth algorithm [1].

Relation (5) for $Q = 2$ becomes

$$PR_i = \frac{1}{4} PR_{i-1} + \left(-y_{N-2i} + \frac{1}{2} y_{N-2i+1} + \frac{1}{2} y_{N-2i+2}\right)X, \qquad i = 1,2,\cdots,N/2$$

and represents a modified form of the "uniform shift of two" algorithm given in [2] and [3]. An advantage of the form given here is that $N/2$ complete steps are exactly required, without the need of a final shift.

Relation (5) for $Q = 3$ becomes

$$PR_i = \frac{1}{8} PR_{i-1} + \left(-y_{N-3i} + \frac{1}{2} y_{N-3i+1} + \frac{1}{4} y_{N-3i+2} + \frac{1}{4} y_{N-3i+3}\right)X, \qquad i = 1,2,\cdots,N/3$$

and represents a modified form of the "uniform shift of three" algorithm described in [2].

Likewise, from relation (3) the following uniform shift of Q multiplication algorithm results to be defined:

$$PL_i = PL_{i-1} + z(Q)_{i-1} X_{i-1}, \quad X_i = 2^{-Q} X_{i-1}, \qquad i = 1,2,\cdots,M, \quad (6)$$

where $PL_0 = 0$ and $X_0 = X$. The ith step of the algorithm is as follows:

1) the multiplier bits are scanned from the left to the right in order to construct $z(Q)_{i-1}$;

2) the quantity X_{i-1} (initially $X_0 = X$) first is multiplied by $z(Q)_{i-1}$ and is then added to the previous partial product (initially the partial product is zero), so forming a new partial product;

3) the quantity X_{i-1} is arithmetically right shifted Q places, so obtaining X_i.

It should be noted that for the left- to right-scanning algorithm an adder and registers of double precision are required.

References

[1] A. D. Booth, "A signed binary multiplication technique," *Quart. J. Mech. Appl. Math.*, vol. 4, pt 2, 1951.
[2] O. L. MacSorley, "High-speed arithmetic in binary computers," *Proc. IRE*, vol. 49, pp. 67–91, Jan. 1961.
[3] L. P. Rubinfield, "A proof of the modified Booth's algorithm for multiplication," *IEEE Trans. Comput.*, vol. C-24, pp. 1014–1015, Oct. 1975.
[4] G. W. Reitwiesner, "Binary arithmetic," *Advances in Computers*, vol. 1, pp. 231–308, 1960.
[5] J. O. Penhollow, "A study of arithmetic recoding with applications to multiplication and division," Rep. 128, Dep. of Comput. Sci., University of Illinois, Urbana, IL, Sept. 1962.
[6] H. L. Garner, "Number systems and arithmetic," *Advances in Computers*, vol. 6, pp. 131–194, 1965.

The Automatic Counting of Asbestos Fibers in Air Samples

T. PAVLIDIS AND K. STEIGLITZ

***Abstract*—A method is described for automating the counting of asbestos fibers in air samples by computer processing of digitized pictures. Preliminary results show the method is feasible.**

***Index Terms*—Asbestos, asbestos fibers, counting, fibers, image encoding, image processing.**

I. Introduction

Presently, the levels of hazardous asbestos fibers in certain industrial environments are monitored by human counting from magnified air samples. In this paper we propose a system for accomplishing the same end automatically, by digitizing the microscope image and using a computer program to count the fibers. The results of some preliminary runs are presented to show the ultimate feasibility of such a scheme.

The advantages of automating a monitoring function such as that discussed here are obvious: first, there is the consistency and reliability inherent in an automatic process; second, the speed; third, the ultimate decrease in cost.

The practical implementation of an automatic fiber counting system can be accomplished by either local or remote computing. If local computing is done, there is no communication problem, but the capital cost of a dedicated minicomputer must be considered. If remote computing is done, a large time-shared system can be used, but the communication costs must be considered. The choice between these approaches will be governed mostly by economic considerations. In either case the microscope picture can be digitized by one of the many available techniques. In the present work we digitized photographic prints (Fig. 1) using a TV camera, a scan rate converter and an A-D converter into 256 × 256 pixels with 6 bits/pixel.

Manuscript received April 25, 1977. This research was supported by NSF Grants ENG72-04133, GK-42048, PHS Grant GM-20289, and U.S. Army Research-Durham under Contract DAAG29-75-G-0192.

The authors are with the Department of Electrical Engineering and Computer Science, Princeton University, Princeton, NJ 08540.

II. Data Structure and Basic Algorithm

Before any further processing each digitized picture is converted into a graph. This offers an immediate data compaction and simplifies the subsequent processing steps [1], [2]. The conversion is done in the following way: Each raster is scanned for dark areas, i.e., where the brightness is less than some predefined threshold T. When a "dark" interval is found it is assigned a number and it is considered as a node of the graph. If two "dark" intervals in adjacent lines overlap then the corresponding nodes of the graph are connected by a branch. Overlap of two intervals is defined if they have at least a pair of cells one directly above the other [Fig. 2(a)]. An alternative criterion is to define overlap if they have a pair of cells with a common corner. [Fig. 2(b)]. We used the first method in the present implementation. The graph is directed through the above-below relation of the intervals corresponding to nodes.

Obviously fiber ends will be mapped into nodes of total degree one [Fig. 2(c)]. However, nodes of total degree one can also occur from bent or split fibers as shown in Fig. 3(a). This configuration will yield a connected component of exactly two nodes. However, the arrangement of Fig. 3(b) yields three nodes of degree one as shown in Fig. 3(d). A multiply bent fiber [Fig. 3(c)] will give a graph of the form shown in Fig. 3(e). On the other hand, two crossing fibers will give the configurations of Fig. 3(f) or (g). It can be seen that any node of total degree one from a bent fiber must be the start of a "downward" path to a node of degree (1, 2) (denotes up-degree = 1, down-degree = 2) or an upward path to a node of degree (2,1) or to a node of degree one. On the other hand, crossing fibers can generate only paths from nodes of degree (0,1) to (2,1) and (2,2) or from nodes of degree (1,0) to (1,2) and (2,2). If the graph is searched and all nodes of degree (0,1) or (1,0) connected by a chain of nodes of degree (1,1) to nodes of degree (1,2) (or (2,1) respectively) are marked then the unmarked nodes of total degree 1 will correspond exactly to fiber ends. Therefore, the number of fibers will equal half the number of such nodes.

The arguments above are based on the assumption that the quantization width is small compared with the thickness of the fibers.

The last assumption seems to hold in most practical situations and the actual number of fibers is uncertain enough so that the above algorithm can be considered as a heuristic. In fact, inspection of Fig. 1 shows that the main problem is the breaking of fibers by nonuniform illumination, resulting in small segments comparable in size to the quantization width.

Obviously further work on more sophisticated algorithms will result in more accurate counts, and the present algorithm is primarily aimed at demonstrating the feasibility of the approach.

III. Implementation

The procedures described in the previous section were implemented in Fortran and tested on a number of pictures digitized by the method described in Section I. Figure 4 is the digitization of that part of Fig. 1 which is within dotted lines. It is seen that the 256 × 256 matrix does not give high enough resolution for all the available detail. However, it was decided to proceed with these data for two reasons. One was economical and the other had to do with the application. Fibers missed by the 256 × 256 quantization will be of diameter less than 0.2μ and this is below the generally accepted limit of those constituting a health hazard [3].

Fig. 5 shows the graph produced by the algorithm. The numbers indicate the correspondence of nodes of degrees (0,1), (1,0), (0,2), (2,0), and (0,0) with Fig. 4. Nodes #82 and #139 are marked. The count produced by the algorithm is then 8 which is close to a human count on the original picture. (Note that it is difficult if not impossible for human observers to agree on how