On-Line Algorithms for Division and Multiplication*

Kishor S. Trivedi
Department of Computer Science
Duke University
Durham, N.C., 27706

and

Milos D. Ercegovac
Computer Science Department
University of California
Los Angeles, California 90024

*This work was carried out when the authors were at the University of Illinois at Urbana-Champaign and was supported in part by the National Science Foundation under Grant No. US NSF DCR 73-07998.

1. Introduction

In this paper we are considering problems of division and multiplication in a computational environment in which all basic arithmetic algorithms satisfy "on-line" property: to generate j^{th} digit of the result it is necessary and sufficient to have argument(s) available up to the $(j+\delta)$th digit, where the index difference δ is a small positive constant. Such an environment, due to its potential to perform a sequence of operations in an overlapped fashion, could conveniently speed up an arithmetic multiprocessor structure or it could be useful in certain real-time applications, with inherent on-line properties. The on-line property implies a left-to-right digit-by-digit type of algorithm and consequently, a redundant representation, at least, of the results. For addition and subtraction such algorithms, satisfying on-line property, can be easily specified. Multiplication requires a somewhat more elaborate approach and there are several possible ways of defining an on-line algorithm. However, the existence of an on-line division algorithm is not obvious and its analysis appears interesting.

After introductory remarks, an analysis of the on-line division problem is given in Section 2. The radix-2 case with non-redundant operands is considered first and a feasible on-line algorithm is defined. Later, a generalization of this on-line algorithm for redundantly represented operands and higher radices is given. In Section 3, a compatible on-line multiplication algorithm is considered, while in the concluding section certain aspects of implementation and performance are discussed.

2. Division

Let us denote the dividend, the divisor and the quotient by N, D and Q respectively, such that $N = \sum_{i=1}^{m} n_i r^{-i}$, $D = \sum_{i=1}^{m} d_i r^{-i}$, $Q = \sum_{i=1}^{m} q_i r^{-i}$, and $Q = N/D$ to m digit precision.

If all the digits of the operands N and D are known in advance then the division can be carried out by the following well known algorithm D1.

Algorithm D1:

Step 1 [Initialize]: $P_0 \leftarrow N$; $j \leftarrow 0$;

Step 2 [Selection]: $q_{j+1} \leftarrow \text{Select } (rP_j, D)$;

Step 3 [Basic Recursion]:

$$P_{j+1} \leftarrow rP_j - q_{j+1}D;$$

Step 4 [Test]: If $j < m-1$ then $j \leftarrow j+1$; and go to step 2;

End D1;

P_j is known as the $j^{\underline{th}}$ partial remainder and rP_j is known as the $j^{\underline{th}}$ shifted partial remainder. The procedure Select obtains the new quotient digit q_{j+1} such that P_{j+1} satisfies certain range restrictions [1]. This process requires the comparison of rP_j against some constant multiples of D. The use of redundancy in the representation of each digital position of the quotient allows us to select q_{j+1} based on the inspection of a limited number of (leading) digits of rP_j and D[1]. Furthermore, these methods can also be extended to the case when both the partial remainder and the divisor are in redundant form [2]. It is clear that without the use of redundancy it is not possible to avoid a set of full precision comparisons.

The present problem is that the digits of the dividend and the divisor are not known in advance but are available on line, digit by digit, most significant digit first. It should be clear at the outset that in the absence of redundancy the problem can not be solved.

Let us assume that the first digit (q_1) of the quotient can be obtained after δ leading digits of the dividend and the divisor are known. Thereafter one new digit of the quotient can be obtained upon receiving one digit each of the dividend and the divisor. We can then specify the following algorithm D2.

Algorithm D2:

Step 1 [Initialize]: $\hat{P}_0 \leftarrow \sum_{i=1}^{\delta} n_i r^{-i}$;

$\hat{D}_0 \leftarrow \sum_{i=1}^{\delta} d_i r^{-i}$; $j \leftarrow 0$;

Step 2 [Selection]: $q_{j+1} \leftarrow \text{Select1 } (r\hat{P}_j, \hat{D}_j)$;

Step 3 [Basic Recursions]:

$$\hat{D}_{j+1} \leftarrow \hat{D}_j + d_{j+\delta+1} r^{-j-\delta-1}$$

$$\hat{P}_{j+1} \leftarrow r\hat{P}_j + n_{j+\delta+1} r^{-\delta} - q_{j+1}\hat{D}_{j+1} - \left(\sum_{i=1}^{j} q_i r^{-i}\right) \cdot d_{j+\delta+1} \cdot r^{-\delta}$$

Step 4 [Test]:

If $j < m-1$ then $j \leftarrow j+1$ and go to step 2;

End D2;

Note that, we have assumed that the dividend and the divisor are padded with δ zero digits to the right.

From the recursions of algorithm D2, we have,

$$\hat{P}_j = r^j\left[\sum_{i=1}^{j+\delta} n_i r^{-i} - \left(\sum_{i=1}^{j} q_i r^{-i}\right)\left(\sum_{i=1}^{j+\delta} d_i r^{-i}\right)\right] \quad (2.1)$$

which implies,

$$\hat{P}_m = r^m[\,N - QD\,].$$

Therefore, if we can devise a selection procedure such that $|\hat{P}_j| \le \alpha|D|$ where $\frac{1}{2} \le \alpha \le 1$ is a small constant then $Q = N/D$ to m digit precision [1]. Instead of doing this directly, we will specify a selection procedure which guarantees that $|P_j| \le \alpha'|D|$ and we will also obtain a bound on $|P_j - \hat{P}_j|$. We will first discuss a simple case in which the radix is two and the divisor and the partial remainders are in nonredundant form. Later, we will generalize to an arbitrary radix and all operands being redundant.

2.1 Binary Division with Nonredundant Operands

From the recursions of algorithm D1, we have,

$$P_j = r^j\left[\sum_{i=1}^{m} n_i r^{-i} - \left(\sum_{i=1}^{j} q_i r^{-i}\right)\left(\sum_{i=1}^{m} d_i r^{-i}\right)\right] \quad (2.2)$$

From equations (2.1) and (2.2), we have,

$$P_j - \hat{P}_j = r^j\left[\sum_{i=j+\delta+1}^{m} n_i r^{-i} - \left(\sum_{i=1}^{j} q_i r^{-i}\right)\left(\sum_{i=j+\delta+1}^{m} d_i r^{-i}\right)\right] \quad (2.3)$$

Note that $n_i \in \{0,1\}$, $d_i \in \{0,1\}$,

$r = 2$, and $q_i \in \{\bar{1},0,1\}$, therefore,

$$P_j - \hat{P}_j \le 2^j[\sum_{i=j+\delta+1}^{m} 2^{-i} + (\sum_{i=1}^{j} 2^{-i})(\sum_{i=j+\delta+1}^{m} 2^{-i})]$$

and

$$P_j - \hat{P}_j \ge -2^j[(\sum_{i=1}^{j} 2^{-i})(\sum_{i=j+\delta+1}^{m} 2^{-i})].$$

From which, we have,

$$-2^{-\delta} \le P_j - \hat{P}_j \le 2 \cdot 2^{-\delta} \qquad (2.4)$$

We will assume that $\frac{1}{2} < D < 1$. From the digit set $\{0,1,\bar{1}\}$ of quotient digits, we obtain the range restriction on P_j,

$$-D < P_j < D \qquad (2.5)$$

From (2.4) and (2.5), we get,

$$-D + 2^{-\delta} < \hat{P}_j < D - 2 \cdot 2^{-\delta} \qquad (2.6)$$

Applying the range restriction (2.5) on P_{j+1}, and using the recursion of algorithm D1, we can determine the range of $2P_j$ for each possible value of q_{j+1}. Thus,

if $0 < 2P_j < 2D$ then $q_{j+1} = 1$,
if $-D < 2P_j < -D$ then $q_{j+1} = 0$, and
if $-2D < 2P_j < 0$ then $q_{j+1} = -1$.

Using inequality (2.4), we get the corresponding selection regions for $2\hat{P}_j$:

If $2 \cdot 2^{-\delta} < 2\hat{P}_j < 2D - 4 \cdot 2^{-\delta}$ then $q_{j+1} = 1$,
if $-D + 2 \cdot 2^{-\delta} < 2\hat{P}_j < D - 4 \cdot 2^{-\delta}$ then $q_{j+1} = 0$,
and if $-2D + 2 \cdot 2^{-\delta} < 2\hat{P}_j < 0 - 4 \cdot 2^{-\delta}$ then $q_{j+1} = -1$.

These conditions can be graphically described by means of a P-D plot [2], as in Figure 2.1. The difference between this P-D plot and the conventional P-D plot is that the ordinate is $2\hat{P}_j$ instead of $2P_j$.

The value of δ is determined by requiring that the lower bound for $q_{j+1} = 1$ region and the upper bound for $q_{j+1} = 0$ region intersect at a value of $D < \frac{1}{2}$. This means $\delta \ge 4$. We choose $\delta = 4$. And two selection lines that we choose are $2\hat{P}_j = \frac{1}{4}$ and $2\hat{P}_j = -\frac{1}{4}$. Thus selection rules are:

If $2\hat{P}_j > \frac{1}{4}$ then $q_{j+1} = 1$,
if $2\hat{P}_j < -\frac{1}{4}$ then $q_{j+1} = -1$,

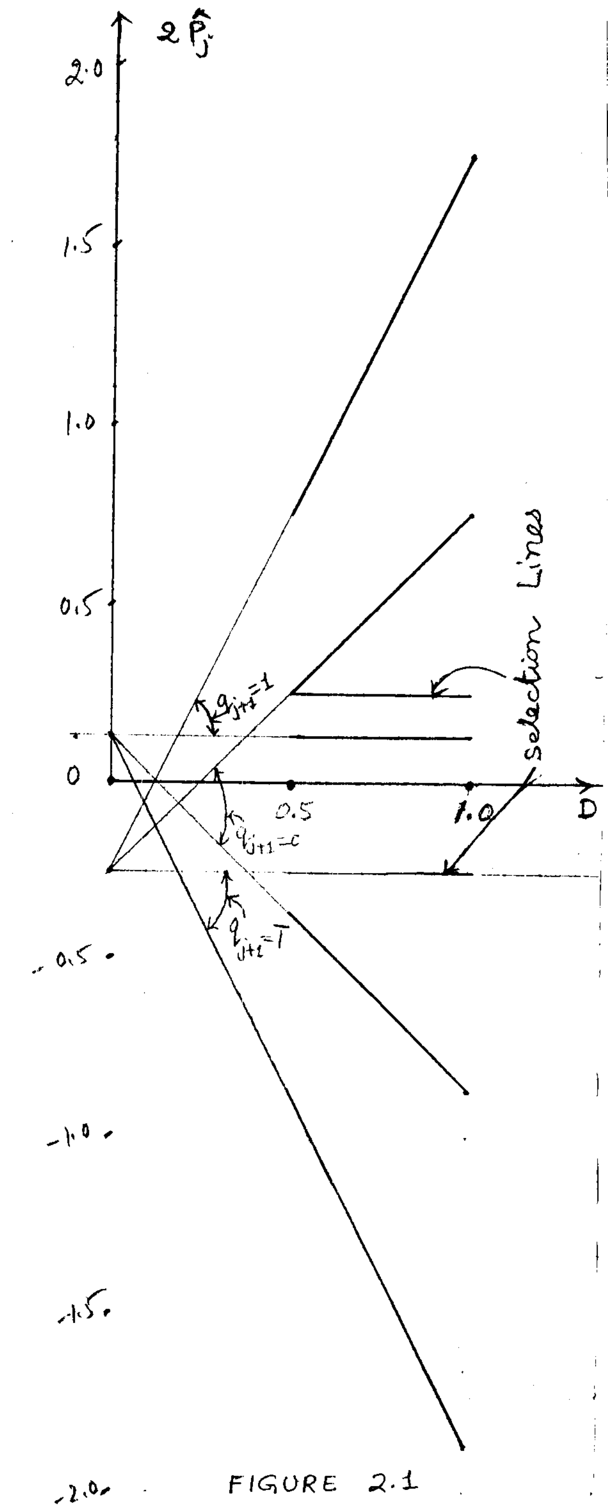

FIGURE 2.1

otherwise

$q_{j+1} = 0$.

To satisfy the range restriction on $\hat{P}_0$, we may have to shift it to the right by at most two bits. Thus we have completed the specification of algorithm D2. It is clear that the same treatment can be extended to higher radix division with nonredundant operands.

An example of the method now follows:

Let $m = 24$,

$N = 0.101000110110101110010101$

and $D = 0.111101100101100011110011$.

$\hat{P}_j$	q_{j+1}
0.101000000000000000000000	1
0.010100000000000000000000	1
1.101001000000000000000000	-1
0.010000010000000000000000	1
1.100111100000000000000000	-1
0.001100100000000000000000	1
1.011100110100000000000000	-1
1.111001100110000000000000	0
1.110011101110000000000000	-1
0.100110010111110000000000	1
0.001111001001100000000000	1
1.100100101101110000000000	-1
0.001011000000100000000000	1
1.011001110001101010000000	-1
1.101110011111100000100000	-1
0.010111111001110000100000	1
1.110011100100000100011000	-1
0.100100101101111010101000	1
0.001111111011000011011000	1
1.011111101110011010100001	-1
1.111101110101011001001000	0
1.111011101010110010010011	0
1.110111010101100100100100	-1
0.101100010000101100111010	1

We note that negative values of $\hat{P}_j$ are represented in two's complement notation.

2.2 Division with Redundant Operands

We assume that the digits of the dividend, the divisor and the quotient are all chosen from the symmetric redundant digit set,

$D_\rho = \{-\rho, \ldots, \bar{1}, 0, 1, \ldots \rho\}$ and $\frac{r}{2} \leq \rho \leq r-1$.

We will denote the degree of redundancy, $\rho/r-1$, by K.

From equation (2.3) we have,

$$|P_j - \hat{P}_j| \leq r^j[\rho \cdot \frac{r^{-j-\delta-1} - r^{-m-1}}{1 - r^{-1}} + \rho \cdot (\frac{r^{-1} - r^{-j-1}}{1 - r^{-1}}) \cdot \rho \cdot (\frac{r^{-j-\delta-1} - r^{-m-1}}{1 - r^{-1}})]$$

$$= (K + K^2)r^{-\delta} - (K + K^2)r^{-m+j} - K^2r^{-j-\delta} + K^2r^{-m}$$

which implies

$$-(K + K^2)r^{-\delta} \leq P_j - \hat{P}_j \leq (K + K^2)r^{-\delta} \qquad (2.7)$$

The range restriction on P_j is given by,

$$-KD < P_j < KD \qquad (2.8)$$

From (2.7) and (2.8), we get the range restriction on $\hat{P}_j$:

$$-KD + (K + K^2)r^{-\delta} < \hat{P}_j < KD - (K + K^2)r^{-\delta} \quad (2.9)$$

Applying the range restriction (2.8) on P_{j+1} and using the recursion of algorithm D1, we can **determine** the **selection** region of rP_j for each possible value of q_{j+1}. Let $q_{j+1} = i$ such that $-\rho \leq i \leq \rho$, then i-selection region is given by,

$$(-K + i)D < rP_j < (K + i)D \qquad (2.10).$$

Corresponding i-selection region for $r\hat{P}_j$ is obtained using (2.7) and (2.10) as,

$$(-K + i)D + (K + K^2)r^{-\delta+1} < r\hat{P}_j < (K + i)D - (K + K^2)r^{-\delta+1} \qquad (2.11)$$

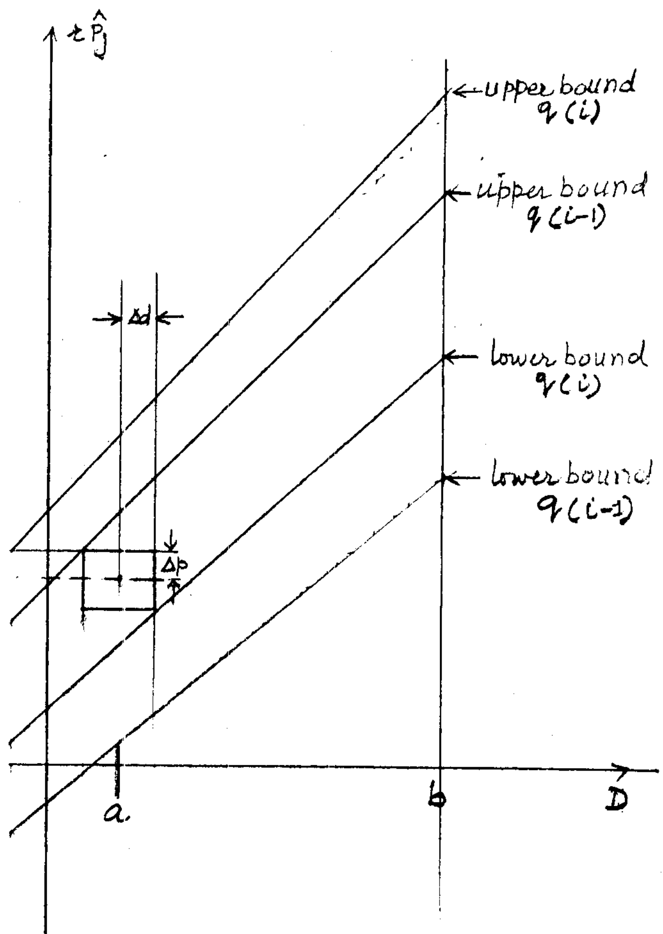

FIGURE 2.2

In Figure 2.2, we have **represented these regions** by means of P-D plot of $r\hat{P}_j$ vs. D. If we want to avoid full precision comparisons of $r\hat{P}_j$ with multiples of D, we have to use the estimates of the full precision $r\hat{P}_j$ and D. Let us assume that we use δ most significant digits of $\hat{P}_j$ as its estimate R_j and β most significant digits of the divisor D as its estimate $\hat{D}$. Note that only $\delta + j$ digits of the divisor are known at step j. Therefore, $\beta \le \delta$. From this point on, we can follow the analysis in Atkins's Ph.D. thesis [2] substituting the P-D plot of $r\hat{P}_j$ vs. D for the original P-D plot. The condition for determining δ and β can now be stated. A rectangle of height $2\Delta p$, width $2\Delta d$ and whose center is at $\hat{D}_{min}$ should be completely contained in the (i,i-1) selection overlap region. Where, Δp and Δd are defined by,

$$|r\hat{P}_j - rR_j| \le \Delta p \quad \text{and}$$
$$|D - \hat{D}| \le \Delta d .$$

The above condition is equivalent to the condition,

$$\Delta p + \tfrac{1}{2}(2i - 1)\,\Delta d + (K + K^2)r^{-\delta+1} \le \tfrac{1}{2}(2K - 1)\hat{D}_{min} .$$

Note that the worst case occurs when $i = \rho$, therefore we get the condition,

$$\Delta p + (\rho - \tfrac{1}{2})\,\Delta d + (K + K^2)r^{-\delta+1} \le (K - \tfrac{1}{2})\hat{D}_{min} \quad (2.12)$$

We can easily obtain,

$$\Delta p \le \frac{\rho}{r-1}\, r^{-\delta+1} \quad \text{and}$$
$$\Delta d \le \frac{\rho}{r-1}\, r^{-\beta} .$$

Therefore, the inequality (2.12) reduces to

$$Kr^{-\delta+1} + (\rho - \tfrac{1}{2})Kr^{-\beta} + (K + K^2)r^{-\delta+1} \le (K - \tfrac{1}{2})\hat{D}_{min}.$$

or

$$(2K + K^2)r^{-\delta+1} + K(\rho - \tfrac{1}{2})r^{-\beta} \le (K - \tfrac{1}{2})\hat{D}_{min}. \quad (2.13)$$

If we assume that the divisor D is standardized, i.e. $d_1 \ne 0$, then

$$\hat{D}_{min} = \frac{1}{r} - \frac{\rho}{r^2}\,\frac{1 - r^{-\beta+1}}{1 - r^{-1}}$$
$$= \frac{1}{r} - \frac{\rho}{r}\,\frac{1 - r^{-\beta+1}}{r - 1}$$
$$= \frac{1}{r}(1 - K(1 - r^{-\beta+1}))$$

Therefore, the condition (2.13) reduces to,

$$(2K + K^2)r^{-\delta+1} + K(\rho - \tfrac{1}{2})r^{-\beta} \le \frac{1}{r}(K - \tfrac{1}{2})[1 - K(1 - r^{-\beta+1})]$$

or,

$$(2K + K^2)r^{-\delta+2} + K(\rho - K)r^{-\beta+1} \le (1 - K)(K - \tfrac{1}{2}) \quad (2.14)$$

Once r, and K are decided, we can determine β and δ from this condition. Note that K = 1, or equivalently maximal redundancy, cannot be used.

As an example, for $r = 10$, $\rho = 6$, and $K = 2/3$, we have $(4/3 + 4/9)10^{-\delta+2} + 2/3(16/3)10^{-\beta+1} \le \frac{1}{3}\cdot\frac{1}{6}$, or

$$3200\ 10^{-\delta} + 640 10^{-\beta} \le 1.$$

If we let $\delta = \beta + 1$ then

$$(320 + 640)10^{-\beta} \le 1 \quad \text{or}$$
$$\beta \ge 3,\ \delta \ge 4.$$

We can continue further and obtain the selection rules but we omit the details here.

3. Multiplication

An on-line algorithm for multiplication, compatible to the previously considered on-line division algorithm, can be conveniently derived following the well-known technique of incremented multiplication, as used in the digital differential analyzers [3,4], combined with the use of redundant number systems.

Let

$$X = \sum_{i=1}^{m} x_i \cdot r^{-i}$$
$$Y = \sum_{i=1}^{m} y_i \cdot r^{-i} \quad (3.1)$$

be the radix r representations of the positive multiplicand and multiplier respectively. Define

$$X^{(j)} = \sum_{i=1}^{j} x_i \cdot r^{-i} = X^{(j-1)} + x_j \cdot r^{-j}$$
$$Y^{(j)} = \sum_{i=1}^{j} y_i \cdot r^{-i} = Y^{(j-1)} + y_j \cdot r^{-j} \quad (3.2)$$

to be the j digit representations of the operands X and Y, available at the j-th step by definition of an on-line algorithm. The corresponding partial product is, then,

$$X^{(j)} \cdot Y^{(j)} = X^{(j-1)} \cdot Y^{(j-1)} + (X^{(j)} \cdot y_j + Y^{(j-1)} \cdot x_j)r^{-j} \quad (3.3)$$

Let $P^{(j)}$ be the scaled partial product, i.e.

$$P^{(j)} = X^{(j)} \cdot Y^{(j)} \cdot r^{j} \quad (3.4)$$

so that the recursion of the multiplication algorithm can be expressed as follows:

$$P^{(j)} = rP^{(j-1)} + X^{(j)}\cdot y_j + Y^{(j-1)}\cdot x_j \qquad (3.5)$$

With $P^{(0)} = 0$, the scaled product $P^{(m)} = X\cdot Y\cdot r^m$ can be generated in m steps (3.5). If a non-redundant number system is used in representing the partial products, the digits of the desired product appear in a right-to-left fashion, as determined by the convential carry propagation requirements. If, however, a redundant number representation system is adopted, the desired left-to-right generation of the product digits, as required by the on-line property, can be easily provided. Moreover, the redundancy in number representation can make the time required to perform the recursive step independent of the operand precision.

Let a symmetric redundant digit set be defined as

$$D_\rho = \{-\rho,-(\rho-1),\dots,-1,0,1,\dots,\rho-1,\rho\} \qquad (3.6)$$

where

$$\frac{r}{2} \le \rho \le r - 1$$

and r is the radix.

Following the general computational method, described in [5], the basic recursion (3.5) of the multiplication algorithm is redefined in the following way:

$$w^{(j)} = r(w^{(j-1)} - d^{(j-1)}) + X^{(j)}\cdot y_j + Y^{(j-1)}\cdot x_j \qquad (3.7)$$

where the digits $\{d_j\}$, $d_j \in D_\rho$, can be determined by the following selection function:

$$d^{(j)} = S(w^{(j)}) = \text{sign } w^{(j)}\cdot\lfloor |w^{(j)}| + \tfrac{1}{2}\rfloor \qquad (3.8)$$

which, clearly, corresponds to a rounding procedure.

Then, by definition,

$$w^{(j)} = P^{(j)} - \sum_{i=1}^{j-1} d^{(i)}\cdot r^{(j-i)} \qquad (3.9)$$

Therefore,

$$P^{(m)} = X\cdot Y\cdot r^m = r^m \sum_{i=1}^{m-1} d^{(i)} r^{-i} + w^{(m)}$$

or $\qquad (3.10)$

$$X\cdot Y = \sum_{i=1}^{m} d^{(i)} r^{-i} + (w^{(m)} - d^{(m)}) r^{-m}$$

By definition of the selection function $S(w^{(j)})$, $|w^{(m)} - d^{(m)}| \le \tfrac{1}{2}$, so that $\sum_{i=1}^{m} d^{(i)} r^{-i}$ is indeed the redundant representation of the most significant half of the product $X\cdot Y$.

In order to preserve the consistency of the recursion (3.7), the bounds on size of the operands must be satisfied. For simplicity, no attempt is made here to establish the necessary operand range conditions, since the following sufficient conditions appear reasonable, simple and practical. The given digit set D_ρ and the selection function $S(w^{(j)})$ imply that

$$|w^{(j)}| < \rho + \tfrac{1}{2} \qquad (3.11)$$

must hold for $j = 1, 2, \dots, m$. Consequently, a sufficient condition on operand bound is

$$|X| < \frac{1}{2r},$$

for a minimally redundant number system with $\rho = \frac{r}{2}$, or,

$$|X| < \tfrac{1}{4}$$

for a maximally redundant number system with $= r - 1$. It can be easily found that these bounds can be improved: for $r = 2$, $|X| \le \tfrac{1}{2}$ is a valid operand range.

As discussed in detail elsewhere [5], it is a simple matter to make the time required for computation of $w^{(j)}$ independent of the precision of the corresponding operands. Namely, by allowing

$$\tfrac{1}{2} < |w^{(j)} - d^{(j)}| < 1$$

a carry propagation free addition can be utilized in (3.7).

The following example illustrates the on-line multiplication algorithm for $r = 2$:

$X = 0.01101001$

$Y = 0.01110011$

j	$X^{(j)}y_j + Y^{(j-1)}x_j$	$w^{(j)}$	$d^{(j)}$	$2(w^{(j)} - d^{(j)})$
1	0.0	0.0	0	0.0
2	0.01	0.01	0	0.1
3	0.101	1.001	1	0.01
4	0.0110	0.1010	1	-0.11
5	0.0111	-0.0101	0	-0.101
6	0.0	-0.101	-1	0.11
7	0.0110100	1.0010100	1	0.010100
8	0.11011011	1.00101011	1	0.0101011

$$\sum_{j=1}^{8} d^{(j)}2^{-j} + 2^{-8}(w^{(8)} - d^{(8)}) = 0.0010111100101011$$

$$= X \cdot Y \quad \square$$

4. Concluding Remarks

Two compatible algorithms for on-line division and multiplication, based on the redundant number systems, have been presented. Besides an obvious usefulness in real-time applications, these on-line algorithms provide an effective way of speeding up the execution of sequences of the basic arithmetic operations by minimizing the delay between successive operations in an overlapped mode of operation. The described algorithms can be seen to have rather simple implementation requirements and properties which are compatible with the desirable modularity in implementation and variable precision operations.

References

[1] Robertson, J. E.: "A New Class of Digital Division Methods," IRE Transaction on Electronic Computers, Vol. EC-7, pp. 218-222, September 1958.

[2] Atkins, D. E.: "A Study of Methods for Selection of Quotient Digits During Digital Division," Ph.D. Dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign, June 1970.

[3] Braun, E. L.: Digital Computer Design-Logic, Circuitry, and Synthesis, Academic Press, New York, 1963.

[4] Campeau T. O.: "Communication and Sequential Problems in the Parallel Processor," in Parallel Processor Systems, Technologies and Applications, Spartan Books, New York, 1970.

[5] Ercegovac, M. D.: "A General Method for Evaluation of Functions and Computations in a Digital Computer," Ph.D. Dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign, July 1975.