THE THEORY AND IMPLEMENTATIONS OF HIGH-RADIX DIVISION

Kwang G. Tan

International Business Machines Corporation
B89/707, PO Box 390
Poughkeepsie, New York 12602

Summary

This report derives the theory of high-radix division in terms of the properties of the overlapped regions of the P-D plot. The minimum precision requirements in quotient selection are discussed. The methods of implementations in hardware and in read-only memory are explored.

Introduction

Division is generally recognized as the most complicated arithmetic operation performed by a computer. In the past twenty-five years, many methods have been developed to increase the speed of the division operation by computer. The process is continuously moving from simple and slow techniques (such as nonrestoring and SRT divisions [1]) towards more complicated but fast algorithms (such as range transformation [2-3] and high-radix divisions [4-6]). In the past, these fast division algorithms were very costly to implement, therefore not widely used in computer design. However, in the LSI technology, circuit counts are no longer the dominant factor in design. High-speed algorithms can be implemented economically to improve the performance of the computer operations. In this report, we shall examine the theory and discuss implementations of a high-radix division algorithm.

Derivation of High-Radix Division

Properties of the P-D Plot

The high-radix division was first proposed by J. E. Robertson [1] and is further studied by D. E. Atkins [4-5]. The fundamental equation for the division iteration is

$$P_{i+1} = rP_i - q_{i+1} \cdot d \tag{1}$$

where

P_i = the partial remainder produced in ith iteration

P_0 = the dividend

d = the divisor

r = the radix

q_{i+1} = the quotient generated for (i+1) ith iteration

It was established by Robertson [1] that the quotient q_i is allowed to be a range of integer values $(-n,-(n-1),\dots -1, 0,1,\dots (n-1),n)$. Clearly $r-1 \geq n$. The above range of integers requires $2n+1$ unique values to represent numbers from 0 to $r-1$, then $2n+1 \geq r$, or $n \geq \frac{1}{2}(r-1)$. Therefore the range of n is determined to be

$$\frac{1}{2}(r-1) \leq n \leq r-1 \tag{2}$$

If the unique values provided by $2n+1$ are more than the r values required for quotient representation $(2n+1 \geq r)$, then we say the quotient is represented redundantly. Define the redundancy constant as $k = n/(r-1)$. Then from Eq. (2):

$$\frac{1}{2} \leq k \leq 1 \tag{3}$$

It is also required that

$$|P_i| \leq k \cdot d \tag{4}$$

Therefore the choice of k for a given r is an important design parameter that must be determined before the actual implementation. Rewrite (1) as

$$rP_i = P_{i+1} + q_{i+1} \cdot d \tag{5}$$

Assume d is bit normalized and is in the range $\frac{1}{2} \leq d < 1$. For a given q_{i+1}, the maximum and minimum rP_i are,

$$(rP_i)_{max} = (k + q_{i+1}) \cdot d \tag{6}$$

$$(rP_i)_{min} = (-k + q_{i+1}) \cdot d \tag{7}$$

CH1412-6/78/0000-C154$00.75

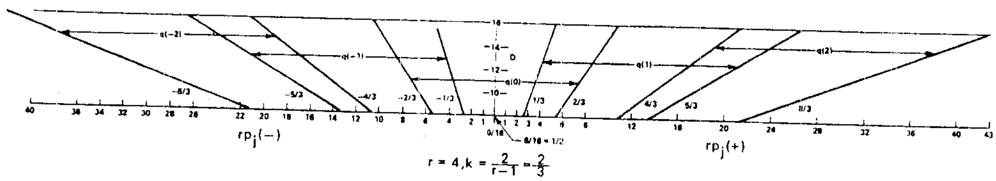

Figure 1a. P-D plot for r=4, k=2/3.

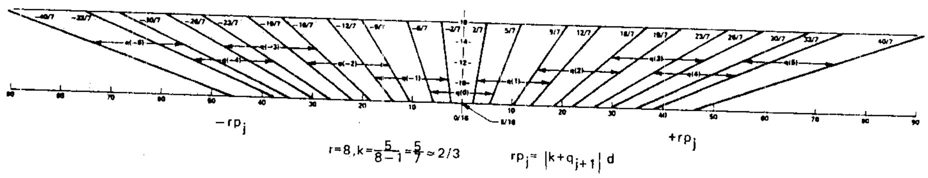

Figure 1b. P-D plot for r=8, k=5/7.

Eq. (5) can then be plotted as a function of d with q_{i+1} as the parameter ranging from -n to n in steps of 1. The area between $(rp_i)_{max}$ and $(rp_i)_{min}$ in which $q_{i+1} = j$ is denoted as the q(j) area. This graphical representation of division iteration is referred to as the P-D plot. The P-D plots for k = 2/3 and r = 4 and r = 8 are shown in Figures 1a and 1b.

The P-D plot will completely specify the division iteration steps. A given divisor d and the shifted partial remainder rp_i will specify a point in a q(j) area. The value j will be the value of the new quotient digit q_{i+1}. In this representation, the redundancy is indicated by the overlapping regions Q(j) between adjacent q(j) and q(j-1) areas, in which either $q_{i+1} = j$ or $q_{i+1} = j-1$ can be the correct choice for the quotient (j range is from 1 to n).

Some General Properties of the Overlapping Regions

The overlapping regions Q(j) play an important part in the quotient selection procedures. Some of its general properties are derived in this section. See Figure 2.

1. The area of each overlapped region Q(j) is developed as follows:

$$\text{Area of } Q(j) = \int_{1/2}^{1} [(rp_{j-1})_{max} - (rp_j)_{min}]dx$$

$$= \int_{1/2}^{1} (2k-1)\ x \cdot dx = 3/8(2k-1) \tag{8}$$

Eq. (8) states that all areas of Q(j) are identical and its size depends on k only. The Q(j) diminishes when $K = \frac{1}{2}$, which corresponds to no redundancy in q.

2. The distance between Q(j) and Q(j+1), the center line of Q(j)

$$= \frac{1}{2}\ [(k+j-1) + (-k+j)]d = \frac{1}{2} \cdot d \cdot (2j-1) \tag{9}$$

The distance between center line of Q(j+1) and Q(j) is

$$\Delta D = \frac{1}{2}\ d\ (2j+1 - (2j-1)) = d \tag{10}$$

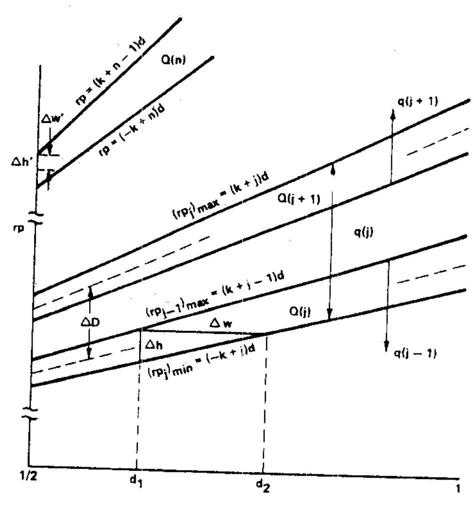

Figure 2. Q(j) regions.

Eqs. (9) and (10) reveal two important facts: (a) The center line of each Q(j) is stationary. It is independent of k. In fact, each Q(j) will expand or shrink symmetrically sbout the center line as k takes on different values. (b) As a consequence of (a), the center-to-center distance ΔD is always equal to the divisor at that point.

3. The height of Q(j)

$$\Delta h = (rp_{j-1})_{max} - (rp_j)_{min}$$

$$= (k+j-1)d - (-k+j)d$$

$$= (2k-1)d \qquad (11)$$

4. To determine the width of Q(j):

$$\Delta w = d_2 - d_1$$

$$= \frac{rp}{-k+j} - \frac{rp}{k+j-1} = rp \cdot \frac{2k-1}{j^2-j+k-k^2} \qquad (12)$$

Quotient Selection in High-Radix Division

The primary advantage of allowing redundancy in the quotient representation is that only the truncated, rather than the full, precision values of rp and d are needed to select the correct quotient digit (q_i) for each iteration. The overlapped region Q(j) plays an important part in the quotient selection process.

Define:

- rp_i, d to be the full precision shifted partial remainder and divisor of m bits each
- $\hat{d}$ to be the truncated divisor
- δ to be the number of bits in $\hat{d}$
- Δd to be the truncated error in $\hat{d}$
  $\Delta d \le 2^{-\delta} - 2^{-m} \approx 2^{-\delta}$ assume m is much larger than δ
- $r\hat{p}_i$ to be the truncated shifted partial remainder
- ϵ to be the number of fractional bits in $r\hat{p}_i$
- Δp to be the truncated error in $r\hat{p}_i$

$$\Delta p \le 2^{-\epsilon} - 2^{-m} \approx 2^{-\epsilon}$$

We have specified $\frac{1}{2} \le d < 1$. Since $d = \hat{d} + \Delta d$, and $d \ge \hat{d}$, therefore $\Delta d \ge 0$. Also $rp_i = r\hat{p}_i + \Delta p$.

a. For $rp_i > 0$, then $rp_i \ge r\hat{p}_i \;\therefore\; \Delta p \ge 0$

b. For $rp_i < 0$, then $r\hat{p}_i \ge rp_i \;\therefore\; \Delta p \le 0$

In both cases, the truncated rp_i will always round the number toward zero.

Since P=D plot is symmetrical about the d-axis, without loss of generality, we shall analyze only the positive regions in which $d > 0$ and $rp_i \ge 0$.

The Truncated Precision Requirement for the Quotient Selection

Inside each overlapped region Q(j), we can select $q_{i+1} = j$ or $q_{i+1} = j-1$ by using the truncated $r\hat{p}_i$ and $\hat{d}$. Outside of each Q(j), the choice is unique for each pair of $(r\hat{p}_i, \hat{d})$. To insure that the quotient selected in Q(j) by the truncated pair $(r\hat{p}_i, \hat{d})$ is always correct, we must require the untruncated pair (rp_i, d) to be located in the same Q(j). In other words, the rectangle with vertices $(r\hat{p}_i, \hat{d})$, $(r\hat{p}_i + \Delta\hat{p}, \hat{d})$, $(r\hat{p}_i, \hat{d} + \Delta d)$, $(r\hat{p}_i + \Delta p, \hat{d} + \Delta d)$ must be completely enclosed within Q(j).

Next we shall locate the region where the maximum truncated precision is required, which corresponds to the region where the height and width of Q(j) are minimum. For a fixed k, from Eq. (11), the minimum height of Q(j) occurs at $d = \frac{1}{2}$. From Eq. (12), the minimum width occurs at $j = n$. Therefore the location in question is situated in Q(n) and close to the rp-axis.

It can be shown that

$$\epsilon \ge 2 - \log_2(2k-1) \qquad (13)$$

$$\delta \ge 2 + \log_2 \frac{(n-k)}{(2k-1)} = 2 + \log_2 \frac{k(r-2)}{(2k-1)} \qquad (14)$$

From Eqs. (13) and (14), ϵ is dependent on k only, while δ is a function of both k and r. Table I shows the minimum ϵ and δ for a range of k and r.

Table I. Minimum precision.

Radix (r)	k = 2/3		k = 3/4		k = 1	
	ε	δ	ε	δ	ε	δ
4	4	4	3	4	2	3
8	4	6	3	6	2	5
16	4	7	3	7	2	6
32	4	8	3	8	2	7
64	4	9	3	9	2	8

A Theoretical Quotient Selection Complexity Measurement

In actual implementation, the usefulness of the overlapped region Q(j), in addition to truncated precision, is to allow the designer some degree of flexibility in choosing the quotient digit so that the hardware required can be minimized. Each Q(j) region is partitioned into a subregion in which q=j and a subregion in which q=j-1; and the subregions are combined in the appropriate q(j) area. The dividing line for the subregions will be a series of staircase steps as indicated in Figure 3.

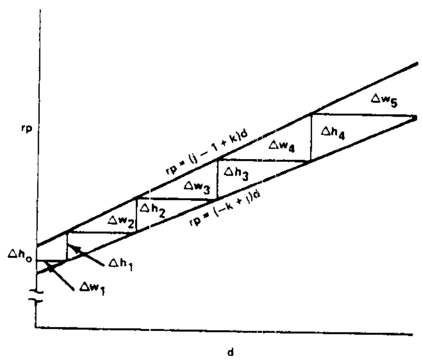

Figure 3. Q region staircase slope.

We shall define and derive the minimum (I) of such steps in Q(j) to be the theoretical measurement of quotient selection complexity.

The overlapped region Q(j) is bounded above by

$$rp = (k+j-1)d \qquad \text{let } v = k+j-1$$
$$= v \cdot d$$

and bounded below by

$$rp = (-k+j)d \qquad \text{let } u = -k+j$$
$$= u \cdot d$$

From Figure 3,

$$\frac{\Delta h_1}{\Delta w_2} = u \qquad \Delta w_2 = \frac{\Delta h_1}{u}$$

$$\frac{\Delta h_2}{\Delta w_2} = v \qquad \Delta h_2 = v \cdot \Delta w_2$$

In general

$$\Delta w_i = \left(\frac{v}{u}\right)^{i-2} \cdot \Delta w_2 \,, \qquad \text{for } i > 2 \tag{15}$$

$$\Delta w_2 = \frac{1}{u} \cdot \Delta h_1 = \frac{1}{u}(\Delta h_0 + v\Delta w_1) = \frac{u+v}{u}\Delta w_1 \tag{16}$$

We want to find the minimum I such that

$$\sum_{i=1}^{I} \Delta w_i \geq 1 - \frac{1}{2} = \frac{1}{2} \tag{17}$$

Let

$$\sigma = \frac{v}{u}\,, \text{ since } v > u\,, \qquad \sigma > 1$$

$$\sum_{i=1}^{I} \Delta w_i = \Delta w_1 + \Delta w_2 \cdot (1 + \sigma \ldots + \sigma^{I-2})$$

$$= [1 + \frac{\sigma+1}{\sigma-1}(\sigma^{I-1} - 1)]\,\Delta w_1 \geq \frac{1}{2}$$

$$\therefore \sigma^{I-1} \geq \frac{1 - 2\,\Delta w_1}{2\,\Delta w_1} \cdot \frac{\sigma-1}{\sigma+1} + 1 \tag{18}$$

$$\frac{\sigma-1}{\sigma+1} = \frac{v-u}{v+u} = \frac{2k-1}{2j-1}$$

From (11) $\Delta h_0 = \frac{1}{2} \cdot \frac{1}{2}(2k-1) = u\,\Delta w_1$

$$\therefore (2k-1) = 4\,u\,\Delta w_1$$

$$\therefore I \geq \frac{1}{\log \sigma} \cdot \log\left(\frac{2(1-2\,\Delta w_1)u}{2j-1} + 1\right) + 1 \tag{19}$$

Let $x = \frac{\sigma-1}{\sigma+1}$, then $\sigma = \frac{1+x}{1-x}$, $0 < x < 1$

$$\log \sigma = \log \frac{1+x}{1-x} = \log(1+x) - \log(1-x)$$

$$= 2[x + \frac{x^3}{3} + \ldots\ldots]$$

$$\approx 2x = \frac{2 \cdot (2k-1)}{2j-1}$$

$$\therefore I \geq \frac{(2j-1)}{2(2k-1)} \cdot \log \left(\frac{2(1-2\,\Delta w_1)u}{2j-1} + 1\right) + 1 \tag{20}$$

From (12), $\Delta w_1 = \frac{1}{2} \cdot \frac{k - \frac{1}{2}}{j-k}$

substitute in (20)

$$I \geq \frac{(2j-1)}{2(2k-1)} \cdot \log\left(2 - \frac{2(2k-1)}{2j-1}\right) + 1 \tag{21}$$

Practical Considerations in Quotient Selection

Determining the Actual Values of Δw_i, Δh_i

In the preceding section, we established the minimum precisions ε, δ required in $r\hat{p}$ and $\hat{d}$. Since we are dealing with truncated precision, the values on the d and rp axes of the P-D plot transform into a set of discrete points. The value of each point on the rp-axis is an integer multiple of $2^{-\varepsilon}$ and on the d-axis is an integer multiple of $2^{-\delta}$. As a consequence, each step of the staircase line that divides Q(j) must also be an integer multiple of the minimum precisions. Let Δw_i, and $\Delta h_i'$ be the actual values in the staircase line, then for each i,

$$\Delta w_i \geq \Delta w_i' = a \cdot 2^{-\delta} , \quad \Delta h_i \geq \Delta h_i' = a' \cdot 2^{-\varepsilon}$$

$$a, a' > 0 \qquad \text{(a, a' are integers)}$$

For example, let r = 16, k = 2/3, n = 10, ε = 4, δ - 7.

$$\Delta h_0 = \frac{1}{2} \cdot \frac{1}{2} \cdot (2 \cdot \frac{2}{3} - 1) = 1/12$$

$$2 \cdot 2^{-4} > \Delta h_0 > 1 \cdot 2^{-4} , \therefore \Delta h_0' = 1 \cdot 2^{-\varepsilon} = 2^{-4}$$

$$\Delta w_1 = \Delta h_0/u = \frac{1}{12} \cdot \frac{1}{n-k} = \frac{1}{12} \cdot \frac{1}{10-2/3} = \frac{1}{112}$$

$$2 \cdot 2^{-7} > \Delta w_1 > 2^{-7} , \quad \therefore \Delta w_1' = 1 \cdot 2^{-\delta} = 2^{-7}$$

When the real values $\Delta w_i'$, $\Delta h_i'$ are used, more steps will be required than the theoretical number of steps calculated from Eq. (20) to traverse the overlapped regions.

The Restrictions on Quotient Selection in Q(j)

Because the points in the overlapped region Q(j) are multiples of the minimum precisions $2^{-\varepsilon}$ and $2^{-\delta}$, and in order to satisfy the criterion stated for truncated precision that $(r\hat{p}_i + \Delta p, \hat{d} + \Delta d)$ be in Q(j), not all the points in Q(j) can be selected for q=j or q=j-1 with complete freedom. For example, in Figure 4, point A $[(r\hat{p}, \hat{d}) = (\frac{12}{16}, \frac{36}{64})]$ is in Q(2). However, point A can not be selected for q=2 because, if the untruncated errors be $\Delta p = 0$, and $\Delta d \approx 2^{-6}$, then

$$rp = r\hat{p} = \frac{12}{16}$$

$$d = \hat{d} + \Delta d = \frac{36}{64} + \frac{1}{64} = \frac{37}{64}$$

Clearly the point $(\frac{12}{16}, \frac{37}{64})$ is in q(1) but not in Q(2). Therefore, point A can be assigned only to q=1. Similarly, point B $= (\frac{25}{16}, \frac{38}{64})$ is in Q(3), but can be assigned only to q=3, not to q=2. Because if the untruncated (rp, d) be

$$rp = r\hat{p} + \Delta p = \frac{25}{16} + \frac{1}{16} = \frac{26}{16}$$

$$d = \hat{d} = \frac{38}{64}$$

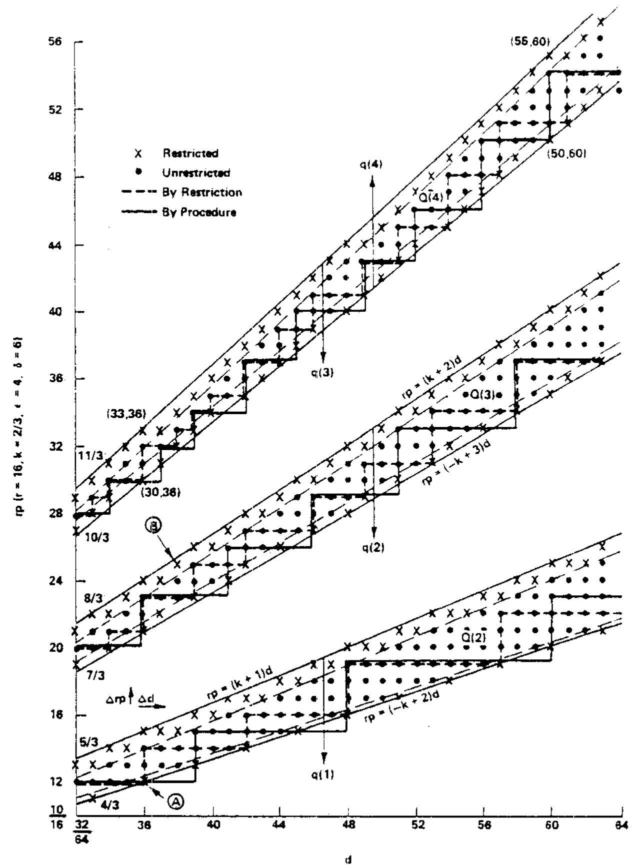

Figure 4. The restricted points in Q(j).

Then the point clearly falls outside of Q(3) and is in q(3).

In fact, all points for which Δp is less than $2^{-\varepsilon}$ distance away from the upper bound of Q(j) can be assigned only to q-j. All points for which Δd is less than $2^{-\delta}$ distance away from the lower bound of Q(j) must be assigned to q=j-1. Only the points on or between the following pair of lines can be assigned to either q=j or q=j-1.

$$\text{inner upper bound} \quad rp = (k + (j-1))d - 2^{-\varepsilon} \qquad (22a)$$

$$\text{inner lower bound} \quad rp = (-k + j)(d + 2^{-\delta}) \qquad (22b)$$

The Selection Procedure for the Minimum Number of Steps

In a previous section, we have established a theoretical quotient selection complexity criterion. It was shown to be a function of the initial step Δw_1. We have also shown that the actual step values that can be used are less than the theoretical values for each step. Hence the actual number of steps may be more than that calculated by Eq. (21). We have further indicated the restrictions placed on quotient selection in $Q(j)$. Our goal in this section is to establish a procedure based on the restrictions derived from previous sections for constructing the steps of the staircase dividing line so that the actual number of steps will be minimized under the stated restrictions and will approach the theoretical minimum as determined by Eq. (21).

Recall Eqs. (22a) and (22b) placed a pair of inner bounds for the points in $Q(j)$, which can be translated into the following selection criteria:

1. The point $(r\hat{p}, \hat{d})$ can be assigned to $q=j$ or $q=j-1$, if the 3 other vertices $(r\hat{p}_i + 2^{-\varepsilon}, \hat{d})$, $(r\hat{p}, \hat{d} + 2^{-\delta})$ and $(r\hat{p} + 2^{-\varepsilon}, \hat{d} + 2^{-\delta})$ are all in $Q(j)$.
2. If all vertex points are not in $Q(j)$, and if any one of them is located in $q(j)$, then $(r\hat{p}, \hat{d})$ must be assigned to $q=j$; or if it is in $q(j-1)$ then $(r\hat{p}, \hat{d})$ must be assigned to $q=j-1$. In conjunction with the above criteria, we present the following procedure for constructing the steps of the dividing line in $Q(j)$:

   a. The initial horizontal step starts out from the center point of each $Q(j)$ on the rp-axis.

   b. The horizontal step line will terminate at the point $(r\hat{p}, \hat{d})$ when the right two vertex points $(r\hat{p}, \hat{d} + 2^{-\delta})$ and $(r\hat{p} + 2^{-\varepsilon}, \hat{d} + 2^{-\delta})$ are outside of the overlapping region $Q(j)$

   c. The vertical step line will start at the terminal point of b. and move up until it reaches a point which is on or just below the inner upper bound line defined by Eq. (22a).

   d. Take the last point in c. and repeat b. and c. above until the right side of the P-D is reached.

   e. All the points on or below the step dividing line are assigned to $q=j-1$ and merged with the $q(j-1)$ area. All points above the step dividing line are assigned to $q=j$ and merged with the $q(j)$ area.

The example in Figure 4 compares the dividing steps constructed by the restrictive bounds, and by the above procedure. A portion of the dividing lines for $r=16$, $k=2/3$ based on the procedure is presented in Figure 5.

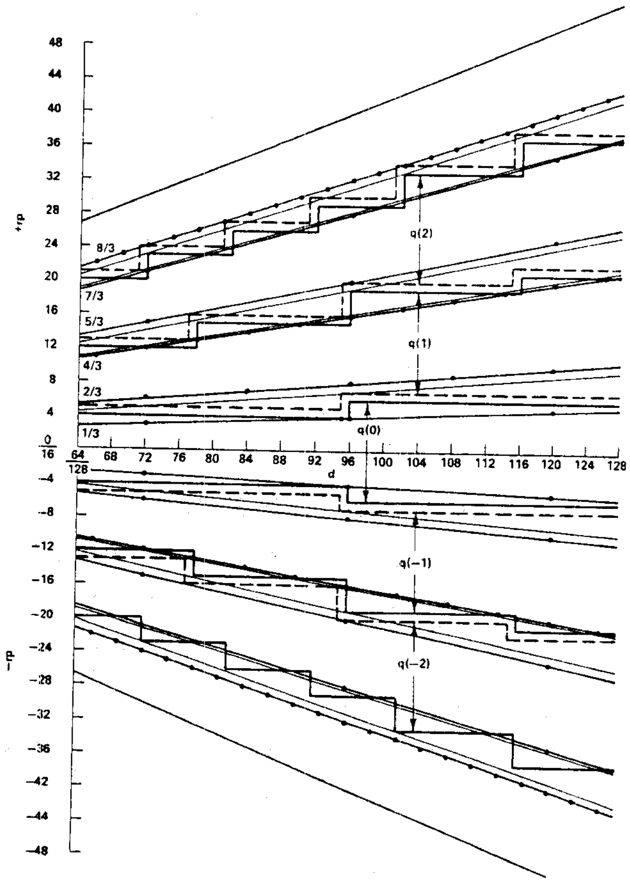

Figure 5. Quotient selection dividing lines for r=16, k=2/3

To indicate the relative complexity of quotient selection discussed in the preceding sections (the number of steps in each $Q(j)$), Table II shows a comparison for the case $r=16$, $k=2/3$, $n=10$, $\varepsilon=4$, and $\delta=7$. The number of steps in each $Q(j)$ is directly proportional to the hardware implementation cost of each $q(j)$ area, as we shall see in the next section.

Table II. The number of steps in the overlapped regions for three different procedures, r=16.

j	k = 1		k = 2/3				
	Δw_1	I from (23)	$\Delta w_1'$ from (16)	$\Delta w_1'$ (actual)	I from (23) Theoretical Min.	I Restricted	I" Derived (procedure)
1	-	1	1/4	1/4	2	2	2
2	1/2	2	1/16	1/16	4	5	4
3	1/8	3	1/28	1/32	6	9	6
4	1/12	3	1/40	3/128	8	14	10
5	1/16	4	1/52	1/64	10	18	12
6	1/20	5	1/64	1/64	12	21	15
7	1/24	5	1/76	1/128	14	28	17
8	1/28	6	1/88	1/128	17	34	22
9	1/32	7	1/100	1/128	19	36	23
10	1/36	8	1/112	1/128	21	39	28

Division Implementation

High-Radix Quotient Selection by Hardware

As noted before, the P-D plot is symmetrical with respect to the d-axis. By using the procedure described earlier, in the negative half of the P-D plot where $r\hat{p} > rp$ and $\Delta p \leq 0$, the dividing staircase line for each of the Q(-j) regions can be made to be the mirror image of its positive counterpart Q(j). After the sign and the magnitude of $r\hat{p}$ are determined, only the positive half of the P-D plot will be sufficient to describe the complete division. (Note: For higher operational speed, it may be necessary to implement all positive and negative q(j) areas.)

Consider the case r+16, k=2/3. The P-D plot with all q(j) area dividing lines is shown in Figure 5. The heart of high-radix division implementation involves the implementation of q(j) areas. The complexity of hardware implementation for each q(j) area directly affects the operating speed of the division iterations. It was shown in the last section that the quotient selection complexity is measured by the number of steps (I) of the dividing line in each Q(j). Since each q(j) area is bounded above and below by the staircase lines, it is possible to subdivide each q(j) area into a number of overlapping rectangles. The total number of rectangles in each q(j) is equal to I_{j+1}. The sum of of all Boolean functions for the rectangles in each q(j) provides the basis of hardware implementation for each quotient selection. As can be expected, the total hardware quotient selection implementation costs are proportional to the sum of all q(j) areas implementation. For r=16, the total hardware costs are substantial.

Therefore, the implementor should utilize the freedom provided in each Q(j) area to construct the best staircase dividing line within the procedure stated above so that the implementation of each q(j) area can be minimized.

There are many hardware organizations in which high-radix division method can be implemented. In general, the system requirement for division speed dictates the radix to be used in divide iteration. Figure 6 indicates a scheme [6] which tends

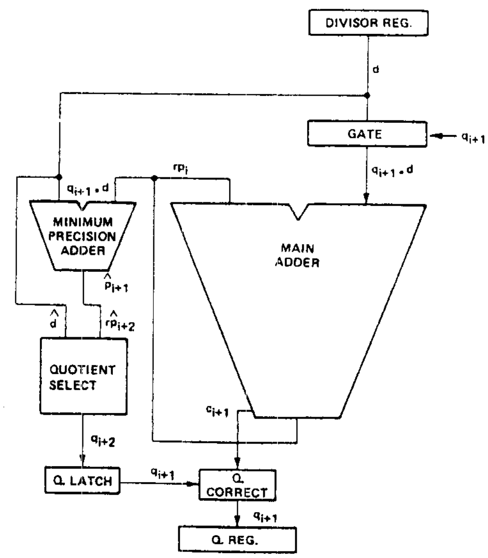

Figure 6. Suggested hardware organization for division implementation.

to minimize the delays of each divide iteration, and the scheme has been implemented in IBM high performance processors. In this scheme, while the main adder is carrying out the full precision divide iteration, the truncated partial remainder (P_{i+1}) is generated in the small minimum precision adder in a parallel operation. The shifted result $(r\hat{p}_{i+2})$ is used to generate the quotient (q_{i+2}) of the next iteration by the procedure defined above while the quotient of current iteration (q_{i+1}) is being corrected by the sign of the iteration.

As suggested in [7], carry-save adders can also be used instead of a propagate adder to further increase the speed of divide iteration. In both schemes, the quotient selections are overlapped with the main full add/subtract operation in order to speed up the divide iteration.

High-Radix Quotient Selection by Control Memory

Quotient Selection by Read-Only Memory (ROM). Since a read-only memory is very economical and its density and speed have been steadily improving, it becomes quite attractive to investigate the organization and the total bit requirements for the quotient selection mechanism (the P-D plot) implemented in the read-only memory.

The organization essentially consists of a two-dimensional array. As shown in Figure 7, $\hat{d}$ is used to address the column

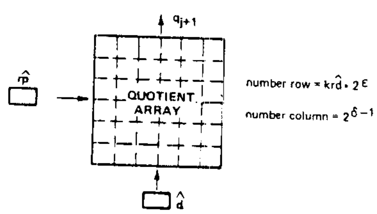

Figure 7. The ROM for quotient selection.

of the array and $r\hat{p}$ addresses the row. The location where these two addresses intersect contains the correct quotient to be used by the division iteration. In other words, the two-dimensional array is the exact replicate of the P-D plot.

The number of rows and columns is determined by the number of distinct values in $r\hat{p}$ and $\hat{d}$ respectively.

Let δ, ε be the minimum precisions for $\hat{d}$ and $r\hat{p}$.

Since $\frac{2^{\delta-1}}{2^{\delta}} = \frac{1}{2} \le \hat{d} \le (1-2^{-\delta}) = \frac{2^{\delta}-1}{2^{\delta}}$

The number of distinct values in $\hat{d}$ is

$$(2^{\delta}-1) - 2^{\delta-1} + 1 = 2^{\delta-1}$$

Since $0 \le r\hat{p} \le kr\hat{d}$, the maximum number of distinct values in $r\hat{p}$ is $(kr\hat{d} \cdot 2^{\varepsilon})$.

The total number of locations (L) in the array is

$$L = \hat{d}_{max} \cdot r\hat{p}_{max}$$

$$= 2^{\delta-1} \cdot (kr\hat{d} \cdot 2^{\varepsilon}) = 2^{\varepsilon} \cdot kr \cdot \frac{(2^{\delta}-1)}{2^{\delta}} \cdot 2^{\delta-1}$$

$$\approx 2^{\varepsilon+\delta-1} \cdot kr \qquad (23)$$

In addition, the binary coded quotient at each location requires $\lceil \log_2 r \rceil$ bits. Therefore the maximum number of bits (B) required is

$$B_{max} = \lceil \log_2 r \rceil \cdot kr \cdot 2^{\varepsilon+\delta-1} \qquad (24)$$

Since no quotient is needed beyond rp = rkd line, the minimum number of bits required is

$$B_{min} = \log_2 r \cdot \sum_{\hat{d}=1/2}^{1} r\hat{p}$$

$$= \log_2 r \cdot \int_{1/2}^{1} rp \cdot d(d) \cdot 2^{\varepsilon} \cdot 2^{\delta}$$

$$= \log_2 r \cdot 2^{\varepsilon+\delta} \cdot \int_{1/2}^{1} rkd \cdot d(d)$$

$$= 3/4 \cdot \log_2 r \cdot 2^{\varepsilon+\delta-1} \cdot rk \qquad (25)$$

Eqs. (24) and (25) provide the upper and lower bound for total bits requirement. Table III tabulates these bounds for different r and k.

Table III. Bits requirement for radix r.

Radix(r)	Total bits (B)					
	k = 2/3		k = 3/4		k = 1	
	Max	Min	Max	Min	Max	Min
4	$2/3 \cdot 2^{10}$*	$1/2 \cdot 2^{10}$	NA	NA	128	96
8	$8 \cdot 2^{10}$	$6 \cdot 2^{10}$	$4.5 \cdot 2^{10}$	$3.375 \cdot 2^{10}$	1536	1152
16	$42.6 \cdot 2^{10}$	$32 \cdot 2^{10}$	$24 \cdot 2^{10}$	$18 \cdot 2^{10}$	$8 \cdot 2^{10}$	$6 \cdot 2^{10}$
32	$213 \cdot 2^{10}$	$160 \cdot 2^{10}$	$120 \cdot 2^{10}$	$90 \cdot 2^{10}$	$40 \cdot 2^{10}$	$30 \cdot 2^{10}$
64	$1024 \cdot 2^{10}$	$768 \cdot 2^{10}$	$576 \cdot 2^{10}$	$432 \cdot 2^{10}$	$192 \cdot 2^{10}$	$144 \cdot 2^{10}$

* 2^{10} = 1K

A Two-Level Control Memory Implementation for Quotient Selection. The total bits obtained in the last section may be too large to operate at a high access rate. The two-level quotient selection scheme shown in Figure 8 will drastically reduce the high speed memory requirement and can lead to very high speed division iteration.

The P-D plot is implemented in the second-level control memory as explained in

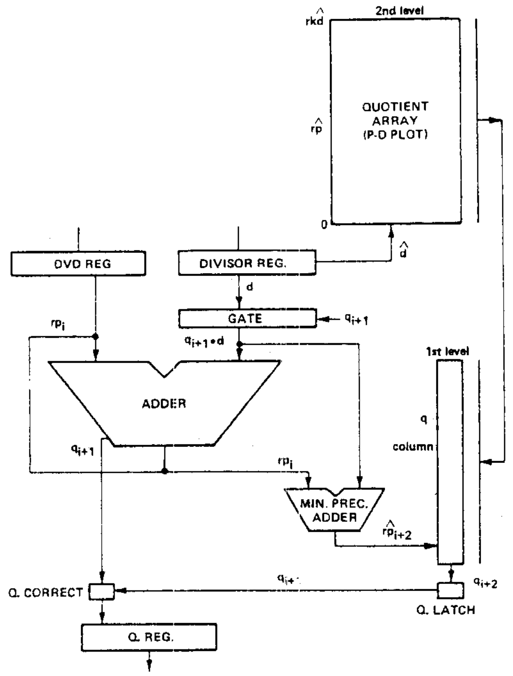

Figure 8. Quotient selection with two-level control store.

the previous section. After a divisor has been normalized, the column of the P-D plot corresponding to $\hat{d}$ is accessed by using the high order bits of the divisor as the address and is gated to a small (only one column of P-D plot) but fast first-level memory or register. Because the divisor d will remain unchanged during the division process, the $r\hat{p}_i$ generated at each iteration is used as address to access the correct quotient bits resided in the first-level memory. Because of its relatively small size, it is possible to make the access time of this memory as fast as the hardware quotient implementations. For a given radix r, the size of the first-level memory is $rk \cdot (\log_2 r) \cdot 2^{\varepsilon}$. For r=4 and k=2/3, it requires only 80 bits.

Using a two-level control memory scheme for quotient selection, it is possible to make the delays of the division iteration independent of the radix. Given a large second-level memory, high-speed radix-16 division iteration can be easily implemented.

Quotient Representations

As indicated before, the quotient generated in each division iteration includes both positive and negative digits. The negative quotient digits must be converted to the final positive form. Also in each iteration, the partial remainder (p_j) can become negative. In such a case, the selected quotient digit must be compensated. The quotient conversion and compensation processes can be performed either at the end of each division iteration or, if the quotient digit is represented redundantly, at the end of the divide operation.

Let q_j' be the final form of the quotient digit of jth iteration. In general, q_j' can be generated by the following rules:

For $q_j \geq 0$
 if $p_j \geq 0$, then $q_j' = q_j$, else $q_j' = q_j - 1$
For $q_j < 0$
 if $p_j < 0$, then $q_j' = r + q_j - 1$, else $q_j' = r + q_j$

Table IV illustrates the generation of q_j' for r=4, k=2/3.

Table IV. Quotient determination

q_j	q_j' $(p_j \geq 0)$	q_j' $(p_j < 0)$
2	2	1
1	1	0
0	0	3
-1	3	2
-2	2	1

If q_j is represented redundantly, it may take the following form:

q_j	$\{^x_y\}$
2	10 11
1	01 11
0	00 11
-1	10 00
-2	01 00

After all divide iterations are completed, the quotient digits (q_j) for all iterations are added in a conventional adder with a low-order one inserted. The results of the addition will be the final quotient digits. The advantage of a redundantly represented quotient is that the quotient digits are corrected automatically. However, additional registers and extra addition operation will be required for this mode of operation.

Conclusion

Theories, methods, and procedures for the implementation of high-radix division have been discussed in detail in this report. It should be clear that both direct hardware implementation and read-only memory implementation of high-radix division are practical in the LSI technology available today.

Acknowledgment

The work was done in 1973 when the author was in Syracuse University under the IBM residence fellowship program.

References

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

2. A. Svoboda, "An Algorithm for Division," Information Processing Machine (Prague, Czech.), No. 9, pp. 25-32, 1963.

3. E. V. Krishnamurthy, "On Range-Transformation Techniques for Division," IEEE Trans. on Computers, Vol. C-19, No. 2, pp. 157-160, February 1970.

4. D. E. Atkins, "Higher Radix Division Using Estimates of the Divisor and Partial Remainders," IEEE Trans. on Computers, Vol. C-17, No. 10, pp. 925-934, October 1968.

5. D. E. Atkins, "A Study of Methods for Selection of Quotient Digits during Digital Division" (Ph.D dissertation), Report No. 397, Dept. of Computer Science, Univ. of Illinois, Urbana, Juen 1970.

6. R. M. Wade, "A Carry-Independent Quarternary Division Scheme," IBM Poughkeepsie Technical Report, TR 00.1531 (September 1966).

7. K. G. Tan, "Uniform 2 Bits Quotient Binary Division by Carry-Save Adders," IBM Technical Disclosure Bulletin, Vol. 14, No. 11, pp. 3279-3281, April 1972.