A Radix-16 SRT Division Unit with Speculation of the Quotient Digits *

Gianluca Cornetta
Computer Architecture Dept.
Universitat Politècnica de Catalunya
08034 Barcelona—Spain
E-mail: cornetta@ac.upc.es

Jordi Cortadella
Software Dept.
Universitat Politècnica de Catalunya
08034 Barcelona—Spain
E-mail: jordic@lsi.upc.es

Abstract

The speed of a divider based on a digit-recurrence algorithm depends mainly on the latency of the quotient digit generation function. In this paper we present an analytical approach that extends the theory developed for standard SRT division and permits to implement division schemes where a simpler function speculates the quotient digit. This leads to division units with shorter cycle time and variable latency since a speculation error may be produced and a post-correction of the quotient may be necessary. We have applied our algorithm to the design of a radix-16 speculative divider for double precision floating point numbers, that resulted to be faster than analogous implementations.

1. Introduction

Most VLSI implementations of the division operation are based on digit recurrence; such class of algorithm, also known as SRT, guarantees a good tradeoff between implementation cost and performance. It is an iterative algorithm with linear convergence, i.e. a digit q_j of the result is calculated at each iteration. The quotient after k iterations is $Q[k] = \sum_{j=1}^{k} q_j \times r^{-j}$, r being the radix. Several techniques proposed to improve the execution time of the algorithm, including, prediction, operands prescaling and overlapping of several selection stages, are reported in the book of Ercegovac and Lang [4]. Speculation of quotient digits, as a technique to speedup the division algorithm, was first reported in [2] and then extended to high-radix division and square root in [3]. The main differences between the method proposed in [2, 3] and the one we have developed are the following:

1. The number of bits for the truncation of the divisor and partial remainder to calculate the quotient digit is analytically derived;
2. The estimations of w and d to be used for speculation, error detection and correction are computed as a function of the desired speculation error;
3. The correction is always carried out in one extra cycle whichever be the error.

The rest of the paper is structured as follows: in section 2 we review the SRT algorithm and introduce the notation that we will use in this paper. Section 3 describes the proposed algorithm. Section 4 applies the theory developed in the previous section to the design of a radix-16 divider and compares the performances with previous implementations. Finally, in section 5, we draw up some conclusions.

2. SRT Division and Notation

In an SRT division scheme, the following recurrence is applied repeatedly:

$$w[j+1] = rw[j] - q_{j+1}d \qquad w[0] = x \tag{1}$$

where $w[j]$ is the residual after the j-th iteration, q_{j+1} the new quotient digit, d is the divisor and x the dividend. We also assume that x and d are positive normalized fractions. The quotient digit is a signed digit $|q_{j+1}| \leq a$ with redundancy factor $\rho = \frac{a}{r-1}$, in particular we will consider digit sets with $\rho \leq 1$. Furthermore, in order to guarantee the convergence of the algorithm, the residual must be bounded, that is:

$$-\rho d \leq w[j] \leq +\rho d \tag{2}$$

The quotient digit q_{j+1} is determined by a digit selection function F whose arguments are an estimation $\hat{w}[j]$ and $\hat{d}$ of both residual and divisor; that is:

$$q_{j+1} = F\left(\hat{w}[j], \hat{d}\right) \tag{3}$$

As described in [4], one of the methods for quotient digit selection is essentially a bound checking operation, i.e. a

*This work has been funded by the Ministry of Education of Spain under CICYT, TIC 98-0410.

comparison of $\hat{w}[j]$ with a set of selection constants m_k that may depend on d and such that $L_k \leq m_k \leq U_{k-1}$, where

$$L_k = (-\rho + k) \times d \qquad U_k = (\rho + k) \times d$$

are respectively the lower and the upper bound of the selection interval for digit $q_{j+1} = k$. A selection function must satisfy two fundamental conditions [4]: *containment* and *continuity*, that determine the range of the selection intervals. However, a speculative selection function must only satisfy the latter, since a correction is performed every time the residual exceeds the correct bounds.

3. Quotient Digit Speculation

In case of speculation of the quotient digit the recurrence for division becomes [3]:

$$w^s[j+1] = rw[j] - q^s_{j+1}d \tag{4}$$

where q^s_{j+1} is the speculated digit and $w^s[j+1]$ the speculated partial remainder. If the speculation is correct, $w[j+1] = w^s[j+1]$ and $q_{j+1} = q^s_{j+1}$. In case of wrong prediction both the partial remainder and the speculated quotient digit have to be corrected. In particular the correct digit $q_{j+1} \in \{q^s_{j+1} - \phi, \ldots, q^s_{j+1} - 1, q^s_{j+1}, q^s_{j+1} + 1, \ldots, q^s_{j+1} + \phi\}$ (with $\phi \geq 1$) results:

$$q_{j+1} = q^s_{j+1} + q^c_{j+1} \tag{5}$$

where $q^c_{j+1} \in \{-\phi, \ldots, -1, 0, +1, \ldots, \phi\}$ is the *correction digit*. The wrong speculated remainder may be corrected using the following relation:

$$w[j+1] = w^s[j+1] - q^c_{j+1} \times d \tag{6}$$

The selection bounds for digit $q^s_{j+1} = k$ are:

$$L^s_k = (-\rho + k - \phi)d \qquad U^s_k = (\rho + k + \phi)d \tag{7}$$

where $\varepsilon = |w[j+1] - w^s[j+1]| = \phi d$ is the maximum speculation error and is a function of d. Since $L^s_k < L_k$ and $U^s_k > U_k$, this results in a larger overlap between two consecutive selection intervals. Consequently the selection function will be simpler. The overlap between the intervals may be increased for values of d close to $\frac{1}{2}$ considering the maximum speculation error for $d = 1$ (namely $\varepsilon = \phi$). Digit q^s_{j+1} is determined by a function F^s that evaluates an estimation of both the full precision residual and divisor, that is:

$$q^s_{j+1} = F^s(w'[j], d') \tag{8}$$

If t is the number of bits of the fractionary part of $\hat{w}[j]$, then the estimation $w'[j]$ of the residual at the j-th iteration may be obtained from $\hat{w}[j]$ by discarding τ least significant bits of its fractionary part. To determine τ we impose that the overlap (for the worst case $d = \frac{1}{2}$) between two adjacent selection intervals must be greater or equal the truncation error of the *carry-sum* redundant representation of $w'[j]$, namely:

$$(U^s_k - L^s_{k+1})_{d=1/2} = (\rho + \phi) - \frac{1}{2} \geq 2^{\tau - t + 1} \tag{9}$$

A carry-save form residual is represented by two bit vectors; the former stores the sum bits whereas the latter the carry bit of an addition. Imposing the *continuity condition* between two consecutive selection intervals ($L^s_k(d_i + 2^{-\sigma}) \leq U^s_{k-1}(d_i)$), we are able to find the number σ of bits of the estimation d' of the divisor, which leads to:

$$(-\rho + \frac{1}{2}) + (-\rho + a - \phi)2^{-\sigma} \leq \phi \tag{10}$$

The term d_i represents the discretized divisor; namely $d_{i+1} = d_i + 2^{-\sigma}$ with $d_0 = \frac{1}{2}$ and $0 \leq i < 2^{-\sigma+1}$.

3.1. Error Detection and Correction

A correct speculation is performed if and only if, at each step j of the recurrence, the speculated partial remainder $w^s[j]$ results to be bounded; that is:

$$-\rho d \leq w^s[j] \leq \rho d \tag{11}$$

Thus, in order to verify the correctness of a prediction we may check if the partial remainder generated by a speculation falls within the correct bounds expressed by (11). Furthermore, the correction function has to guarantee that the *sufficient condition* for convergence (2) be always respected. Since a wrongly speculated partial remainder exceeds the correct bounds expressed by (11) it is surely greater than unity, as a consequence ξ integer bits of the remainder have to be evaluated in order to detect a speculation error. As $\varepsilon = \phi d$, the speculated remainder is included in the following bounds:

$$-(\rho + \phi)d \leq w^s[j] \leq (\rho + \phi)d \tag{12}$$

Since $\rho \leq 1$ and $d < 1$, considering the worst case we obtain:

$$\xi = \lceil \log_2(1 + \phi) \rceil + 1 \qquad \phi \leq r - 1 \tag{13}$$

Which also takes into account of the sign bit.

The speed of the comparison may be augmented by considering the estimations $w''[j]$ and d'' of both $w^s[j]$ and d. Namely $w''[j] = w^s[j] - 2^{-e+1}$ and $d'' = d - 2^{-f}$, which is equivalent to truncate the carry-sum representation of $w''[j]$ to the e-th fractionary bit and d to the f-th fractionary bit; thus we obtain:

$$-\rho d'' \leq w''[j] \leq +\rho d'' - 2^{-e+1} \tag{14}$$

In case $w''[j]$ is out of bounds and close to one of the bounds of (14) (for example the lower bound), we must impose that the correction by $q_j^c = -1$ guarantees the convergence, that is:

$$-\rho d'' + 1 \times d \leq +\rho d'' - 2^{-e+1}$$

from which it follows that the amount of truncation is given by the following equation:

$$(\rho - \frac{1}{2}) \geq 2^{-e+1} + 2^{-f} \tag{15}$$

Expression (14) restricts the range of the allowed residuals expressed by (11). This leads, to be conservative, to perform some unnecessary corrections; moreover, in order to guarantee a single correction cycle we have to extend the correction digit set imposing that $\hat{q}_{j+1}^c = q_{j+1}^c + \alpha$ with $\alpha \in \{-1, 0, +1\}$. The new correction digit $\hat{q}_{j+1}^c$ is generated by an error detection and correction function F^c such that:

$$\hat{q}_{j+1}^c = F^c(w''[j], d'') \tag{16}$$

Simulations have shown that the best performances are obtained for $\phi = 1$, which leads to $\hat{q}_{j+1}^c = \{-2, -1, 0, +1, +2\}$. As a consequence function F^c results to be:

$$\hat{q}_{j+1}^c = \begin{cases} -2 & \text{if } w''[j] \leq -\underline{\Phi} \\ -1 & \text{if } -\underline{\Phi} < w''[j] < -\rho d'' \\ +1 & \text{if } \rho d'' - 2^{-e+1} < w''[j] < +\overline{\Phi} \\ +2 & \text{if } w''[j] \geq \overline{\Phi} \end{cases} \tag{17}$$

The lower bound $\underline{\Phi}$ and the upper bound $\overline{\Phi}$ may be calculated in a straightforward way. To compute this bounds we have to impose that a correction by $\pm 2d$ produces a bounded residual. This means that after a correction, the residual $w[j]$ is such that $|w[j]| \leq \rho'' d$. Thus we obtain:

$$\underline{\Phi} = (\rho - 2)d'' - 2^{-f+1} - 2^{-e+1} \tag{18}$$

and

$$\overline{\Phi} = (2 - \rho)d'' - 2^{-f+1} \tag{19}$$

3.2. Performance Evaluation

Performance evaluation of a design is based on the calculation of the average number of cycles per quotient digit (C_d) and of the delay per quotient bit (D_b). Let N_d be the number of divisions simulated, N_c the number of correction cycles that have been performed and m the size of the mantissa (in bits); the number of cycles per quotient digit may be defined as [3]:

$$C_d = 1 + \frac{N_c}{N_d \times \lceil m / \log_2 r \rceil} \tag{20}$$

If D is the *cycle delay* of the design, the delay per bit may be defined as follows:

$$D_b = \frac{C_d \times D}{\log_2 r} \tag{21}$$

4. Implementation and Timing

We have performed the design of the radix-16 unit implementing our algorithm using a standard-cell library (ES2-ECPD10) and SIS for logic minimization. The characteristics of both our design and those presented in [3] are reported in Table 1. Delays and area are expressed as a multiple of the delay and area of a two-input NAND gate with a fanout of three NAND gates. Figure 1 shows the implementation

Radix	16	16 [3]	512 [3]
a	12	12	320
# of CSAs	2	2	4
# bits of w' (sum-carry)	(6,6)	(6,5)	(12, 12)
# bits of d'	2	1	6
# bits of w'' (ξ, e)	(2,4)	(3,5)	(6, 4)
# bits of d''	2	4	3
cycle/digit	1.1	1.3	1.8
cycle delay	28.4	28.8	43.6
delay/bit	8	9.4	8.8
latency	416	489	458
area	5215	4900	8400

Table 1. Characteristics of the implementations.

of the speculative radix-16 divider with $\rho = \frac{12}{15}$. The use of a redundant digit set permits to perform carry-free additions using carry-save adders (CSA). Quotient conversion from redundant to non-redundant form may be performed on the fly, i.e. in parallel with the division operation, as also described in [4]. The dashed line indicates the critical path. In order to reduce the number of CSAs necessary to implement all the multiples of d, digits $q_{j+1}^s \in \{-11, +11\}$ are not speculated since they would need three CSAs, consequently they are obtained exploiting the correction function. Figure 1 shows the cycle time of the divider. In order to speedup the circuit, speculation at step j is overlapped with the error detection at step $j-1$. Each digit q_{j+1}^s results to be the sum of two contributions computed by two different combinational blocks; namely:

$$q_{j+1}^s = q_h + q_l$$

Where $q_h \in \{\pm 8, \pm 4, 0\}$ and $q_l \in \{\pm 4, \pm 2, \pm 1, 0\}$. Figure 2 shows how the selection logic is implemented in case of radix 16 in detail. The residual estimation in carry-save form is first assimilated by a carry-propagation adder (CPA) whose outputs are fed into two different combinational blocks; the former computes q_h and only needs the five most significant bits of the assimilation, the latter computes q_l and needs the whole assimilation. As a consequence the selection logic has a variable delay. Computation of digit q_h

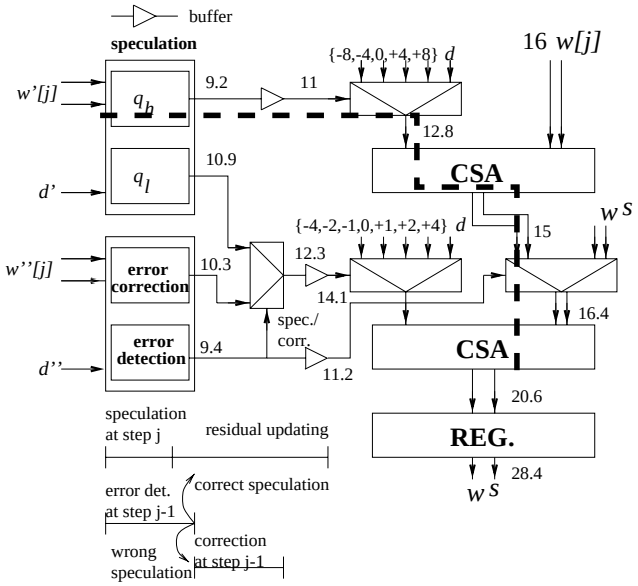

Figure 1. Implementation and Timing of the Radix-16 Speculative Divider.

can be performed faster, due to the reduced inputs, in order to shorten the critical path. On the other hand, computing q_l is a slower operation and is overlapped to q_h computation and to part of the delay introduced by the first CSA in the chain (see Figure 2). What leads to improved performance with respect to the implementations without partial advance described in [3] is the different arrangement of the divisor multiples along the adder chain. The choice of the multiples is determined according to the frequency with which a digit is selected. Due to this consideration, the choice of speculating digits $q^s_{j+1} \in \{-12, +12\}$, unlike [3] where such digits are generated exploiting the correction logic, leads to hit-ratios very close to 90% while the architecture proposed in [3] has a hit-ratio of 70%. In addition, although using one more bit to speculate, we achieve cycles delays comparable to those of [3] designing the selection logic as reported in Figure 2 (since the table that computes q_h requires a number of bits smaller than the assimilation) and using *boolean relations* [5] to synthesize the tables. Also the correction function contributes to speedup the execution time of the architecture since, unlike [3], it always needs only a single cycle to correct an error.

5. Conclusions

Quotient digit speculation is an alternative to conventional division techniques. Assimilating a reduced number of bits of both residual and divisor may lead to a fast selection function and hence to a speedup of the execution time. Nevertheless the speculated digit may be incorrect. The correctness of the selection is checked by an error detection function that verifies whether the speculated residual falls within the correct bounds or not. In case of incorrect speculation the algorithm rolls back and the digit is corrected. This operation is performed by an error correction function. Because of the possible rollbacks, the execution time is variable, i.e. we can have different cycle times for each iteration. As a consequence we have transformed a fixed-latency unit into a variable latency one running with a faster clock cycle and with a functioning that reminds that of the telescopic units introduced in [1]. Moreover, due to its variable latency, our algorithm results particularly appealing for asynchronous implementations too.

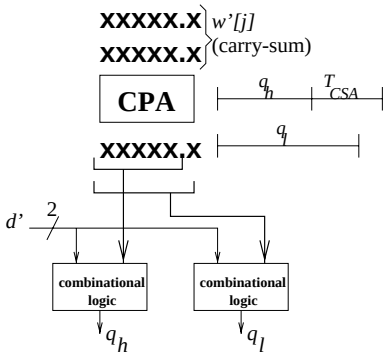

Figure 2. Implementation and Timing of the Radix-16 Selection Function.

References

[1] L. Benini, E. Macii, and M. Poncino. Telescopic Units: Increasing the Average Throughput of Pipelined Designs by Adaptative Latency Control. In *34th Design Automation Conference*, 1997.

[2] J. Cortadella and T. Lang. Division with Speculation of Quotient Digits. In *11th Symposium on Computer Arithmetic*, pages 87–94, 1993.

[3] J. Cortadella and T. Lang. High-Radix Division and Square Root with Speculation. *IEEE Transaction on Computers*, C-43(8):919–931, August 1994.

[4] M.D. Ercegovac and T. Lang. *Division and Square Root. Digit-Recurrence Algorithms and Implementations*. Kluwer Academic Publishers, Norwell, MA, 1994.

[5] Y. Watanabe and R. E. Bryant. Heuristic Minimization of Multipled-Valued Relations. *IEEE Transaction on Computer-Aided Design of Integrated Circuits*, 12(10):1458–1472, October 1993.