A Digit-by-Digit Algorithm for mth Root Extraction

Paolo Montuschi, *Senior Member*, *IEEE*, Javier D. Bruguera, *Member*, *IEEE*, Luigi Ciminiera, *Member*, *IEEE*, and José-Alejandro Piñeiro

Abstract—A general digit-recurrence algorithm for the computation of the mth root (with an m integer) is presented in this paper. Based on the concept of completing the mth root, a detailed analysis of the convergence conditions is performed and iteration-independent digit-selection rules are obtained for any radix and redundant digit set. A radix-2 version for mth rooting is also studied, together with closed formulas for both the digit selection rules and the number of bits required to perform correct selections.

Index Terms—Integer rooting, digit-by-digit algorithms, higher radix, computer arithmetic.

1 Introduction

The design of special-purpose units for the extraction of powers has always been a challenging and stimulating task since the birth of computer arithmetic. In the beginning, the computation of powers and roots was done either by passing through the cascaded chain logarithm-multiplication/division-exponential (L-M/D-E) or by applying the Newton method of tangents to derive a function which, once computed, provided the result. However, as the need for results that are accurate to the last digit has increased (as pointed out by the IEEE 754 standard), the research was directed both to the refinement (and speedup) of these techniques and to the derivation of suitable methods and techniques that are able to provide results that are virtually correct for any required precision. In addition to the improvements of the L-M/D-E class of methods, two main approaches have characterized the scene: the application of general methods for function evaluation[1] to the case of power extraction and the study of special-purpose algorithms and circuits. Extensive work on methods of the first group can be found in [4], [5], [11], [15], and [16]. Concerning the last group, the first power extraction case studied was square root, mainly because it is both more frequent (with respect to other power extractions) and relatively easy to implement. A lot of work has been done on square root: Algorithms for any radix and digit set have been proposed [8], [9] and similarities between SRT-square root and division computation have been demonstrated, resulting in the design of special-purpose units with a combined implementation of division and square root [1], [3], [12].

As both the need for and the offer of more computer power has increased, the horizon has also expanded itself to consider more root extraction cases other than simple square rooting. Good examples showing that, nowadays, research is still focused on the problem of computing root powers are scientific computing, digital signal processing, multimedia, 3D graphics, and so forth. Above all, the works in [13], and [17] are good examples of the natural consequence of such an interest. In particular, [17] presents a general digit-recurrence algorithm and a block-level scheme of an architecture for cube rooting, with particular attention to the algorithm for the radix-2 cube root. On the other hand, [13] presents a general higher radix algorithm/architecture for the computation of the powering function X^Y. This method belongs to the family of L-M/D-E, where, above all, the computations are sped up by using redundancy and online arithmetic.

In this paper, we extend the square root computation by addressing the problem of computing the general mth root (with m integer) for any choice of radix and digit set. We provide a general digit-by-digit algorithm from which the derivation of the algorithms for square root [2], [8], [9] and for cube root [17] can be considered as special cases of the proposed method. Implementations of architectures for SRT digit-by-digit radix-2 cube root are presented in [14], [17]. Because of the properties of radix-2 (as already seen for square rooting), we have dedicated an additional section for studying the characteristics of radix-2. The results are very interesting and intriguing, since radix-2 mth rooting shows unexpected mathematical properties such as closed formulas for the digit selection rules and the number of bits required by these.

Certainly, two strong motivations of this study are to show the mathematical feasibility and related requirements of digit-by-digit mth rooting and to provide the researcher with the results of the most general design framework for

1. Usually, these methods are based on the use of tables plus other circuitry, such as adders and multipliers.

- *P. Montuschi and L. Ciminiera are with the Department of Control and Computer Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino 10129, Italy. E-mail: {paolo.montuschi, luigi.ciminiera}@polito.it.*
- *J.D. Bruguera is with the Department of Electronic and Computer Engineering, University of Santiago de Compostela, Santiago de Compostela, Spain. E-mail: bruguera@dec.usc.es.*
- *J.-A. Piñeiro is with the Intel Barcelona Research Center, Intel Labs-UPC, Barcelona, Spain. E-mail: alex.pineiro@intel.com.*

Manuscript received 30 Jan. 2007; revised 8 May 2007; accepted 14 June 2007; published online 28 June 2007.
Recommended for acceptance by F. Lombardi.
For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-0042-0107.
Digital Object Identifier no. 10.1109/TC.2007.70764.

0018-9340/07/$25.00 © 2007 IEEE Published by the IEEE Computer Society

TABLE 1
Definitions and Symbols

Symbol	Definition	Symbol	Definition
m	root degree, $(m \in N^+, m \geq 2)$	r	radix; $r = 2^b$, $(b \in N^+)$
p	iteration index, $(p \in N^+)$	i	number of computed digits of the root
x	radicand	$w[i]$	residual after the computation of i digits, $(w[0] = x)$
$S[i]$	partial result on i digits $(S[0] = 0)$	n	number of bits of the final result
$S = S[n]$	final result provided by the algorithm	D_a	redundant digit set $= \{-a, \ldots, 0, \ldots, a\}$, $(a \in N^+)$
ρ	index of redundancy; $\rho = a/(r-1)$	s_i	$i-$th result digit; $s_i \in D_a$
$\widehat{w[i-1]}$	estimate of the residual $w[i-1]$	$\widehat{S}$	estimate of the result S (obtained through lookup)
k	possible value of a digit selection	c	number of integer bits of $\widehat{w[i-1]}$
δ	number of fractional bits of $\widehat{S}$	$select(.)$	digit selection function
$g_{[k]}$	constants used to implement $select(.)$	$\widehat{x}$	estimate of the radicand x used by the lookup
v	possible value of $\widehat{S}$	h_v	constants used to implement the lookup of $\widehat{S}$
t	number of fractional bits of $w[i-1]$ originating the estimate $\widehat{w[i-1]}$	u	number of fractional bits of x originating the estimate $\widehat{x}$

digit-by-digit integer mth root. This could be used either for the design of a special-purpose unit or for comparison purposes. Digit-by-digit algorithms have the advantage, with respect to other techniques, of directly providing a result that is correct to the last bit that has been computed. For this reason, the proposed algorithms find their domain of existence where it is (or it will be) necessary to have a precise result of the computations and/or in the cases where intermediate rounding errors between atomic floating-point operations must be avoided [17]. In addition, even neglecting the results from the future development of algorithms requiring the computation of the mth root, nowadays, there exist a number of practical applications where an mth rooting, with $m > 2$, is necessary. As cited in [17], cube rooting is used in the numerical calculation for solving third or fourth-degree algebraic equations and also in computer graphics for color processing. Computer graphics can also require even higher order root computations for volume shading, atmospheric models, and radiance and luminance computations, which use nonlinear models [10].

This paper is organized as follows: In Sections 2 and 3, we first introduce the problem of mth rooting and present the proposed algorithm. Section 4 discusses the problem of lookup for initialization, while Section 5 is specifically dedicated to the analysis of radix-2 mth rooting. To improve the readability, the proofs of the theorems have been moved to the Appendix.

2 Preliminaries

2.1 Square Rooting

Digit-by-digit methods have the characteristic of producing a new digit on the completion of each iteration. For example, it is well known [8], [9] that the algorithm for square rooting in radix-2 is based on the concept of completing the square and, with reference to the symbols defined in Table 1, is regulated by the following recurrence:

$$w[i] = 2w[i-1] - (2S[i-1] + s_i 2^{-i})s_i = (x - S[i]^2)2^i, \quad (1)$$

with $S[i] = S[i-1] + s_i 2^{-i}$. In order to produce a result S that is normalized, that is, $1/2 \leq S < 1$, it is necessary to consider for square root $1/4 \leq x < 1$. Moreover, a convenient choice is to have the residuals $w[i-1]$ in carry-save representation since this implies a simpler and faster updating hardware.

The determination of the digit s_i is carried out by inspecting the value of the residual $w[i-1]$. The use of a redundant digit set, that is, $s_i \in \{-1, 0, +1\}$, avoids full-length comparisons and then allows the estimate $\widehat{w[i-1]}$ to be considered in place of the full-precision residual $w[i-1]$. Therefore, for the digit selection procedure, it is demonstrated [8], [9] that it is necessary to consider the estimate $\widehat{w[i-1]}$ obtained from the four most significant bit positions of the carry-save representation of $w[i-1]$ (that is, including up to weight 2^{-1}). In this case, the digit selection rules [8] are

$$\begin{aligned} s_i &= +1 \quad if \quad 0 \leq \widehat{w[i-1]} \leq 3/2, \\ s_i &= 0 \quad if \quad \widehat{w[i-1]} = -1/2, \\ s_i &= -1 \quad if \quad -5/2 \leq \widehat{w[i-1]} \leq -1. \end{aligned} \quad (2)$$

As the computations continue, it is possible on-the-fly to convert [6] the partially developed square root values $S[i]$ into nonredundant form in order to have the conventional representation of the final square root value S ready after just one additional iteration.

2.2 Extension to mth Root Extraction

In this paper, we extend the SRT square rooting algorithms presented in [2] and the cubic root algorithm in [17] to the more general case of mth root extraction with m integer greater than 1 and we use the notation in Table 1. In order to produce a result S that is normalized, that is, $1/2 \leq S < 1$, we will consider, for the mth root, $2^{-m} \leq x < 1$. For our purposes, we assume x to be available at the beginning of the operations in full precision and in nonredundant representation, thus exluding from this analysis the online algorithms. As for the square root algorithms in [2], we consider the residuals represented in carry-save form. We show that mth root extraction requires a lookup step whose role is to provide a starting value of the result, which allows it to have digit selection rules that are independent of the iteration number. Then, the next digit selection occurs by means of rules based on the lookup value and on the value

of an estimate of the current residual. Our analysis is rather general since we consider any radix r and any redundant digit set with index of redundancy $\rho = a/(r-1) \leq 1$. The extension to overredundant digit sets comes directly from our analysis.

3 The Algorithm

3.1 Recurrence for mth Root Extraction

Theorem 1. *The recurrence for the mth root extraction in base $r = 2^b$ is based on the concept of completing the mth power and is*

$$w[i] = rw[i-1] + r^i[S[i-1]^m - S[i]^m] = (x - S[i]^m)r^i, \quad (3)$$

with $s_i \in D_a$ and $S[i] = S[i-1] + s_i r^{-i}$.

Observe that it is possible to substitute $S[i] = S[i-1] + s_i r^{-i}$ in (3) and to expand the term $S[i]^m = (S[i-1] + s_i r^{-i})^m$ so as to obtain an expression of degree $m-1$ in $S[i-1]$ and s_i.

Based on the recurrence defined by Theorem 1, the key point of the algorithm is to determine the digit selection rules that permit obtaining the digit s_i by inspecting some bits of an estimate $w[\widehat{i-1}]$ of the residual $w[i-1]$, together with the value of the first lookup estimate of the result.

3.2 Region of Convergence and Intervals for Digit Selection

We define the maximum possible domain of the residuals $w[i-1]$ as the region of convergence which still permits the algorithm to work properly [9].

Theorem 2. *The region of convergence of the algorithm based on (3) is given by*

$$\begin{aligned}[-S[i-1]^m + (S[i-1] - \rho r^{-(i-1)})^m]r^{i-1} \leq w[i-1] \\ \leq [-S[i-1]^m + (S[i-1] + \rho r^{-(i-1)})^m]r^{i-1}.\end{aligned} \quad (4)$$

The intervals for digit selection are computed by imposing that the next residual belongs to the region of convergence.

Theorem 3. *The intervals for digit selection $s_i = k$ are*

$$\begin{aligned}[-S[i-1]^m + (S[i-1] + r^{-i}(k-\rho))^m]r^{i-1} \leq w[i-1] \\ \leq [-S[i-1]^m + (S[i-1] + r^{-i}(k+\rho))^m]r^{i-1}.\end{aligned} \quad (5)$$

It can be observed that the selection intervals defined by (5) have their bounds depending on i (also including $S[i-1]$). In order to have fixed digit selection intervals (and rules), it is necessary to eliminate these dependencies from (5). In particular, except in specific cases [7], this requires an initialization of the result before the iterations can start, which is based on the process of digit selection and of updating of the residual. We first address the problem of initialization and then pass to the determination of iteration-independent digit selection intervals.

3.3 Initialization

For $r = 2$, Theorem 10 will show that the initialization phase is trivial. On the other hand, for $r > 2$, we consider the whole mth root extraction, as carried out in two different phases. The first is an initialization where an estimate $\widehat{S}$ of the result is obtained on δ fractional bits, which is then followed by normal iterations based on (3) and on the iteration index p. It is assumed that $p = 0$ during the initialization phase, which implies that $i = \delta/b + p$. Since δ is not necessarily an integer multiple of b, i may be a rational number. However, $b \cdot i$, which is the number of bits produced, is always integer.[2] In the rest of this paper, the algorithms will be analyzed by only referring to the index i since this choice allows a homogeneous notation to be used for both phases.

Theorem 4. *The necessary and sufficient range for the result $\widehat{S}$ provided by the initialization phase, which lets the algorithm converge to the correct result, is*

$$\lfloor 1/2 + \rho 2^{-\delta} \rfloor \leq \widehat{S} \leq \lceil 1 - \rho 2^{-\delta} - 2^{-n} \rceil, \quad (6)$$

where δ is the number of fractional bits of $\widehat{S}$.

It is observed that, for a maximally redundant digit set, that is, $\rho = 1$, (6) becomes

$$1/2 + 2^{-\delta} \leq \widehat{S} \leq 1 - 2^{-\delta}. \quad (7)$$

On the other hand, for $\rho < 1$, we have $1/2 \leq \widehat{S} \leq 1$. The number of bits δ to be produced during the initialization phase will be determined in the next sections for $r > 2$. The case $r = 2$ will be explicitly studied in Section 5.

3.4 Iteration-Independent Digit Selection Intervals

Now, we derive conservative intervals with (5) which do not depend on i. In order to do this, we also have to replace the terms $S[i-1]$ with conservative expressions on $\widehat{S}$. Two possibilities exist: $\widehat{S}$ remains fixed at the value provided by the initialization or $\widehat{S}$ (after the initialization) is dynamically obtained by the first δ fractional bits of $S[i-1]$. In the following, we consider $\widehat{S}$ fixed, the extension to the other case being straightforward. From the definition of $\widehat{S}$,

$$\begin{aligned}\max[1/2, \widehat{S} - \rho(2^{-\delta} - r^{-(i-1)})] \leq S[i-1] \\ \leq \min[1, \widehat{S} + \rho(2^{-\delta} - r^{-(i-1)})].\end{aligned} \quad (8)$$

Theorem 5. *The most conservative intervals for the digit selections $s_i = k$ derived from (5), whose bounds do not depend on $i \geq \delta/b + 1$ and when the initialization is performed, providing $\widehat{S}$ on δ fractional bits, are*

$$\begin{array}{lll} \textit{for } k > 0 & \textit{if} & \frac{m(k-\rho)}{r}[\min(\widehat{S} + \rho 2^{-\delta}, 1)]^{m-1} \leq \\ & w[i-1] & \leq \frac{m(k+\rho)}{r}[\max(\widehat{S} - \rho 2^{-\delta}, 2^{-1})]^{m-1}, \\ \textit{for } k = 0 & \textit{if} & -\frac{m\rho}{r}[\max(\widehat{S} - \rho 2^{-\delta}, 2^{-1})]^{m-1} \leq \\ & w[i-1] & \leq \frac{m\rho}{r}[\max(\widehat{S} - \rho 2^{-\delta}, 2^{-1})]^{m-1}, \\ \textit{for } k < 0 & \textit{if} & \frac{m(k-\rho)}{r}[\max(\widehat{S} - \rho 2^{-\delta}, 2^{-1})]^{m-1} \leq \\ & w[i-1] & \leq \frac{m(k+\rho)}{r}[\min(\widehat{S} + \rho 2^{-\delta}, 1)]^{m-1}. \end{array} \quad (9)$$

2. Observe that the noninteger values for i are caused by the initialization phase, which introduces a fixed shift in both the result and the residual and does not affect, from a practical point of view, the subsequent operations.

3.5 Domain of Existence of the Residual

Let us determine from (4) the tight and conservative domain of $w[i-1]$ not depending on i.

Theorem 6. *The tight domain of existence of $w[i-1]$ derived from the region of convergence (4), whose bounds do not depend on $i \geq \delta/b+1$, is defined as $-L_W < w[i-1] < U_W$, where*

$$\begin{aligned} &\text{for } \rho < 1 \qquad -\rho m < w[i-1] \leq (-1+(1+\rho 2^{-\delta})^m)2^{\delta}, \\ &\text{for } \rho = 1 \qquad -m < w[i-1] < m. \end{aligned} \tag{10}$$

3.6 Digit Selection Rules

It can be observed that the digit selection intervals (9) depend on the residual $w[i-1]$, which is expressed in full precision. However, it is known [8], [9] that it is possible to use the redundancy introduced by considering the digit set D_a to reduce the inspection of $w[i-1]$ to only a limited number of bits, that is, the estimate $\widehat{w[i-1]}$ obtained by truncation up to the tth fractional weight position, with

$$0 \leq w[i-1] - \widehat{w[i-1]} < 2^{-t+1}. \tag{11}$$

Now, for $r > 2$, since $r = 2$ is studied in Section 5, we derive the digit selection rules based on $\widehat{w[i-1]}$ which are used to determine the digits of the result. For division and square root, it is known [8] that this problem is equivalent to the problem of computing a set of constants $g_{[k]}$ such that it is possible to write the digit selection rules as

$$select \quad s_i = k \quad if \quad g_{[k]} \leq \widehat{w[i-1]} < g_{[k+1]}. \tag{12}$$

In the case of carry-save representation of the residual $w[i-1]$, it is known [8] that it is enough for the constants $g_{[k]}$ to satisfy the necessary (but not sufficient) relation

$$L_{[k]} \leq g_{[k]} \leq U_{[k-1]} - 2^{-t}, \tag{13}$$

where $L_{[k]}$ and $U_{[k]}$ define the lower and upper bounds of the domain of $w[i-1]$ related to the generic digit selection $s_i = k$ given by (9), that is (for $1/2 < \widehat{S} < 1$, for example),

$$\begin{aligned} L_{[k]} &= \frac{m(k-\rho)}{r}(\widehat{S}+\rho 2^{-\delta})^{m-1}, \\ U_{[k-1]} &= \frac{m(k-1+\rho)}{r}\cdot(\widehat{S}-\rho 2^{-\delta})^{m-1}. \end{aligned} \tag{14}$$

Theorem 7. *In order to have correct digit selection rules, in the case of the carry-save representation of $w[i-1]$, a necessary (but not sufficient) condition is to both have an estimate $\widehat{w[i-1]}$ of $w[i-1]$ up to its tth fractional bit and consider an initial value $\widehat{S}$ on δ fractional bits given by*

$$\begin{aligned} &\frac{m(\rho r-1)}{r}(2^{-1}+(1-\rho)2^{-\delta})^{m-1} \\ &-\frac{m\rho(r-2)}{r}(2^{-1}+(1+\rho)2^{-\delta})^{m-1} - 2^{-t} \geq 0, \end{aligned} \tag{15}$$

that is, with

$$\begin{aligned} t \geq \log_2\{r/\{m[(\rho r-1)(2^{-1}+(1-\rho)2^{-\delta})^{m-1} \\ -\rho(r-2)(2^{-1}+(1+\rho)2^{-\delta})^{m-1}]\}\}, \end{aligned} \tag{16}$$

$$\delta > \log_2 \frac{2(1+\rho-(1-\rho)[(\rho r-1)/(\rho r-2\rho)]^{1/(m-1)})}{[(\rho r-1)/(\rho r-2\rho)]^{1/(m-1)}-1}. \tag{17}$$

3.7 Number of Bits of $\widehat{w[i-1]}$

Note that, in addition to the t fractional bits, it is necessary to inspect all of the integer bits of $\widehat{w[i-1]}$. From (11) and the region of convergence given by Theorem 6, the domain of $\widehat{w[i-1]}$ results in

$$-L_W - 2^{-t+1} < \widehat{w[i-1]} < U_W. \tag{18}$$

In order to represent all of the values of $\widehat{w[i-1]}$ within the range (18), about $c = \log_2(L_W + U_W + 2^{-t+1})$ integer bits, plus t fractional bits, are necessary.

3.8 Algorithm for Determining the Digit Selection Rules

The determination of the digit selection rules is well known [8] and is carried out as follows:

1. Choose a pair of values t and δ satisfying the necessary conditions of Theorem 7.
2. Determine the constants $g_{[k]}$ according to (9) and (14).
3. Determine the digit selection rules according to (12).

When it is not possible to determine valid digit selection rules (that is, when, in general, it is not true that $g_{[k]} \geq g_{[k+1]}$), it is necessary to operate with another choice of t and δ.

3.9 Avoidance of Explicit Computation of the Powering of $S[i-1]$

The computation of the powering of $S[i-1]$ required by the recurrence (3) can be avoided, provided that a suitable number of additional recurrences is carried out together with (3). Let us define $C_j[i-1] = S[i-1]^j$, with $C_0[i-1] = 1$ and $C_1[i-1] = S[i-1]$. We observe that

$$\begin{aligned} C_p[i] &= (S[i-1] + r^{-i}s_i)^p \\ &= S[i-1]^p + \sum_{j=1}^{p}\binom{p}{j}S[i-1]^{p-j}(r^{-i}s_i)^j \\ &= C_p[i-1] + \sum_{j=1}^{p}\binom{p}{j}C_{p-j}[i-1](r^{-i}s_i)^j. \end{aligned} \tag{19}$$

Observe that, in (19), the computation of $C_p[i]$ only depends on values $C_j[i-1]$ (with $j \leq p$), that is, the values available after the end of the previous iteration $i-1$ plus the shifted powers of s_i.

4 Lookup for Initialization

As seen in Section 3.3, the mth rooting computation requires an estimate $\widehat{S}$ for $r > 2$. We assume the computation of the values v of the estimate $\widehat{S}$ by means of a lookup

table, which is entered by the estimate $\widehat{x}$ obtained by considering the radicand x up to its uth fractional bit.

Theorem 8. *The lookup table intervals are*

$$lookup\ \widehat{S} = v\ if\ (v - \rho 2^{-\delta})^m \leq x \leq (v + \rho 2^{-\delta})^m. \quad (20)$$

Since $\widehat{S}$ is expressed on δ fractional bits, "consecutive" values of $\widehat{S}$ differ by $2^{-\delta}$. Let us denote with $L_{[v]}$ and $U_{[v]}$ the lower and upper bounds of (20). The problem of determining valid lookup table rules based on $\widehat{x}$ is equivalent to the computation of a set of constants h_v that guarantee

$$lookup\ \widehat{S} = v\ if\ h_v \leq \widehat{x} < h_{v+2^{-\delta}}. \quad (21)$$

In the case of nonredundant x, it is known [8] that the constants h_v have to satisfy the sufficient (but not necessary) relation

$$L_{[v]} \leq h_v \leq U_{[v-2^{-\delta}]} - 2^{-u}. \quad (22)$$

Theorem 9. *In order to have correct lookup table rules, a sufficient (but not necessary) condition is to have an estimate $\widehat{x}$ of x up to its uth fractional bit, which is given by*

$$u \geq -\log_2[(2^{-1} - (1-\rho)2^{-\delta})^m - (2^{-1} - \rho 2^{-\delta})^m]. \quad (23)$$

The technique for devising the lookup constants is then similar to that for the determination of the constants $g_{[k]}$ outlined in Section 3.8.

5 Radix-2 mth Root

As stated in the previous sections, radix $r = 2$ is worth a separate analysis since a number of interesting properties that are worth studying exist.

5.1 Starting Value and Digit Selection Intervals

Theorem 10. *When $r = 2$, it is possible to start the iterations for the mth root extraction process by considering $i = 3$, $\widehat{S} = S[2] = 2^{-1} + 2^{-2}$, and $w[2] = (x - S[2]^m)2^2$.*

As a direct consequence of Theorem 10, there is Theorem 11.

Theorem 11. *For $r = 2$, the digit selection intervals are*

$$\begin{aligned} s_i = 1 \quad &if \quad 0 \leq w[i-1] < m, \\ s_i = 0 \quad &if \quad -m2^{-m} \leq w[i-1] \leq m2^{-m}, \\ s_i = -1 \quad &if \quad -m < w[i-1] \leq 0. \end{aligned} \quad (24)$$

5.2 Digit Selection Rules and Number of Bits of the Estimate

From the digit selection intervals (24), the next step is to determine the digit selection rules. We start by identifying the estimate $w[\widehat{i-1}]$ of $w[i-1]$.

Theorem 12. *For $r = 2$, it is necessary and sufficient to use the estimate $w[\widehat{i-1}]$ obtained by considering $w[i-1]$ up to its $t = m - \lfloor \log_2(m) \rfloor$ fractional bit in order to obtain valid digit selection rules, which, for a carry-save representation of the residual $w[\widehat{i-1}]$, are*

$$\begin{aligned} s_i = +1 \quad &if \quad 0 \leq w[\widehat{i-1}] \leq m - 2^{-m+f}, \\ s_i = 0 \quad &if \quad w[\widehat{i-1}] = -2^{-m+f}, \\ s_i = -1 \quad &if \quad -m - 2^{-m+f} \leq w[\widehat{i-1}] \leq -2^{-m+f+1}, \end{aligned} \quad (25)$$

with $f = \lfloor \log_2 m \rfloor$.

The problem now is to determine the number of bits that are necessary and sufficient to represent $w[\widehat{i-1}]$.

Theorem 13. *The estimate $w[\widehat{i-1}]$ is obtained by the most significant $m + 2$ bits of $w[i-1]$.*

From (25), it can be observed that the "granularity" of the representation of $w[\widehat{i-1}]$ is 2^{-m+f}.

5.3 Application Examples

The case of square root (that is, $m = 2$) is included in the algorithm that we have presented in Section 5. By doing the proper substitutions, we obtain the same results as in Section 2.1 and [9].

5.3.1 Radix-2 Cube Root

Our algorithm also applies to the analysis of cube root (that is, $m = 3$). From Theorem 1, we have that, for cube root, the recurrence is

$$w[i] = 2w[i-1] - [3S[i-1]^2 + 3S[i-1]s_i 2^{-i} + s_i^2 2^{-2i}]s_i. \quad (26)$$

From Theorem 12, we get $t = 2$ and the digit selection rules $select\ s_i = +1 \quad if \quad 0 \leq w[\widehat{i-1}] \leq 11/4$, $select\ s_i = 0 \quad if \quad w[\widehat{i-1}] = -1/4$, and $select\ s_i = -1 \quad if \quad -13/4 \leq w[\widehat{i-1}] \leq -1/2$.

Some implementations of architectures for SRT digit-by-digit radix-2 cube root are also presented in [14], [17].

5.3.2 Radix-2 Quadric Root

From Theorems 1, 12, and 13, we get $t = 2$, $w[i-1]$ truncated to its 6 most significant bits (four integer plus two fractional). The recurrence is $w[i] = 2w[i-1] - [4S[i-1]^3 + 6S[i-1]^2 s_i 2^{-i} + 4S[i-1]^2 s_i^2 2^{-2i} + s_i^3 2^{-3i}]s_i$, while the digit selection rules are $select\ s_i = +1\ if\ 0 \leq w[\widehat{i-1}] \leq 15/4$, $select\ s_i = 0 \quad if \quad w[\widehat{i-1}] = -1/4$, and $select\ s_i = -1 \quad if \quad -15/4 \leq w[\widehat{i-1}] \leq -1/2$.

6 Concluding Remarks

In this paper, we have presented a general radix-r digit-recurrence algorithm for the computation of the mth root (with an m integer). We have proposed expressions and techniques for devising valid digit selection and lookup tables which allow the algorithm to converge. We have also considered in detail a radix-2 version for mth rooting and we have found that it has several interesting properties, including the existence of closed formulas for both the digit selection rules and the number of bits required for performing correct selections. With this research, we have shown both the mathematical feasibility and the related requirements of digit-by-digit mth rooting. In addition, we have provided the results of the most general design

framework for digit-by-digit integer mth root, which could be used either for the design of a special-purpose unit or for comparison with future architectures.

APPENDIX

PROOFS OF THEOREMS

Theorem 1. *The recurrence for the mth root extraction in base $r = 2^b$ is based on the concept of completing the mth power and is*

$$w[i] = rw[i-1] + r^i[S[i-1]^m - S[i]^m] = (x - S[i]^m)r^i, \quad (3)$$

with $s_i \in D_a$ and $S[i] = S[i-1] + s_i r^{-i}$.

Proof of Theorem 1. Let us define the residual $w[i]$ as

$$w[i] = (x - S[i]^m)r^i. \quad (27)$$

Since, from (27), we have $w[i-1] = (x - S[i-1]^m)r^{i-1}$, it follows that $w[i] = rw[i-1] + r^i[S[i-1]^m - S[i]^m]$. □

Theorem 2. *The region of convergence of the algorithm based on (3) is given by*

$$\begin{aligned}&[-S[i-1]^m + (S[i-1] - \rho r^{-(i-1)})^m]r^{i-1} \le w[i-1] \\ &\le [-S[i-1]^m + (S[i-1] + \rho r^{-(i-1)})^m]r^{i-1}.\end{aligned} \quad (4)$$

Proof of Theorem 2. We first prove that if the residual $w[i-1]$ belongs to (4), then $w[i]$ also belongs to (4). Then, we show that (4) both ensures the convergence of the algorithm and expresses the widest region of convergence of the algorithm.

Region (4) ensures that the next residual belongs to the same region. We observe that both bounds of (4) share the property of satisfying the full induction principle, that is, if $\underline{B}_{i-1} \le w[i-1] \le \overline{B}_{i-1}$, then it is also $\underline{B}_i \le w[i] \le \overline{B}_i$. In fact, let us consider $w[i-1] = \underline{B}_{i-1} = [-S[i-1]^m + (S[i-1] - \rho r^{-(i-1)})^m]r^{i-1}$. In such a case, this corresponds to the selection $s_i = -a = -\rho(r-1)$ and, according to (3), we get

$$\begin{aligned}w[i] &= rw[i-1] + r^i[S[i-1]^m - S[i]^m] \\ &= r^i[-S[i-1]^m + (S[i-1] - \rho r^{-(i-1)})^m \\ &\quad + S[i-1]^m - S[i]^m].\end{aligned} \quad (28)$$

Now,

$$S[i] = S[i-1] + s_i r^{-i} = S[i-1] - \rho(r-1)r^{-i}.$$

Then, $S[i-1] - \rho r^{-(i-1)} = S[i] - \rho r^{-i}$ and (28) becomes $w[i] = r^i[-S[i]^m + (S[i] - \rho r^{-i})^m] = \underline{B}_i$. Similar computations can also be carried out for the upper bound of (4) and the selection $s_i = +a$.

Region (4) ensures the convergence of the algorithm. The algorithm converges if $\lim_{i\to\infty} w[i]r^{-i} = 0$ as it is $w[i] = r^i(x - S[i]^m)$. Relation (4) can be rewritten for $w[i]r^{-i}$ as follows:

$$\begin{aligned}[-S[i]^m + (S[i] - \rho r^{-i})^m] &= \rho r^{-i}[-mS[i]^{m-1} + \alpha] \le w[i]r^{-i} \\ &\le [-S[i]^m + (S[i] + \rho r^{-i})^m] = \rho r^{-i}[mS[i]^{m-1} + \beta],\end{aligned}$$

where α and β denote the terms containing higher powers of r^{-i} in the binomial expansion of $(S[i] - \rho r^{-i})^m$ and $(S[i] + \rho r^{-i})^m$, respectively. From their definition, both α and $\beta \to 0$ for $i \to \infty$. By taking the limits for $i \to \infty$ of all the terms, it can be seen that $\lim_{i\to\infty} w[i]r^{-i} = 0$ since $\lim_{i\to\infty} \rho r^{-i}[-mS[i]^{m-1} + \alpha] = 0$ and $\lim_{i\to\infty} \rho r^{-i}[mS[i]^{m-1} + \beta] = 0$.

Region (4) expresses the widest region of convergence of the algorithm. By contradiction, let us consider the term $\gamma_P \Theta_{i-1} \ge 0$ and the "enlarged" upper bound of the region of convergence defined by (29) as follows:

$$w[i-1] = [-S[i-1]^m + (S[i-1] + \rho r^{-(i-1)})^m]r^{i-1} + \gamma_P \Theta_{i-1}. \quad (29)$$

By applying (3) for the minimal digit selection $s_i = -a$, we get $w[i] = r^i[-S[i]^m + (S[i] + \rho r^{-i})^m] + r\gamma_P\Theta_{i-1}$. Since, by hypothesis, $w[i-1]$ belongs to the region of convergence, $w[i]$ does too. Therefore, (29) also applies to $w[i]$ and the values of the coefficient γ_P can be determined. From the equations, we obtain $r\gamma_P\Theta_{i-1} = \gamma_P\Theta_i$, which has the solutions $\gamma_P = undefined\ value$ and $\Theta_{i-1} = r^{i-1}$. A similar approach can also be applied for the negative bound, which is "enlarged" by adding $\gamma_N\Theta_{i-1} \le 0$. In this case, the "enlarged" region of convergence of $w[i]r^{-i}$ becomes

$$\begin{aligned}&[-S[i]^m + (S[i] - \rho r^{-i})^m] + \gamma_N \le w[i]r^{-i} \\ &\le [-S[i]^m + (S[i] + \rho r^{-i})^m] + \gamma_P.\end{aligned} \quad (30)$$

By again taking the limits of all three terms in (30), we obtain $\gamma_N \le \lim_{i\to\infty}(x - S[i]^m) \le \gamma_P$. Only the values $\gamma_P = \gamma_N = 0$ guarantee convergence and this completes the proof. □

Theorem 3. *The intervals for digit selection $s_i = k$ are*

$$\begin{aligned}&[-S[i-1]^m + (S[i-1] + r^{-i}(k-\rho))^m]r^{i-1} \le w[i-1] \\ &\le [-S[i-1]^m + (S[i-1] + r^{-i}(k+\rho))^m]r^{i-1}.\end{aligned} \quad (5)$$

Proof of Theorem 3. We look for the largest value $U_{[k,i-1]}$ of $w[i-1]$ for which the digit selection $s_i = k$ ensures that $w[i]$ belongs to the region of convergence (4). From (3) and (4), we have

$$\begin{aligned}w[i] &= rU_{[k,i-1]} + r^i[S[i-1]^m - S[i]^m] \\ &= [-S[i]^m + (S[i] + \rho r^{-i})^m]r^i.\end{aligned} \quad (31)$$

Since $S[i] = S[i-1] + s_i r^{-i}$, we have $S[i] = S[i-1] + kr^{-i}$ and, hence, (31) becomes

$$U_{[k,i-1]} = [-S[i-1]^m + (S[i-1] + r^{-i}(k+\rho))^m]r^{i-1}.$$

With a similar approach, it is possible to also prove the lower bound of (5). □

Theorem 4. *The necessary and sufficient range for the result $\widehat{S}$ provided by the initialization phase, which lets the algorithm converge to the correct result, is*

$$\lfloor 1/2 + \rho 2^{-\delta} \rfloor \leq \widehat{S} \leq \lceil 1 - \rho 2^{-\delta} - 2^{-n} \rceil, \tag{6}$$

where δ is the number of fractional bits of $\widehat{S}$.

Proof of Theorem 4. For $p = 1$, we get $i = \delta/b + 1$. Remembering (3) and denoting with $\widehat{S}$ the lookup value $S_{\delta/b}$, we have (4) evolving into

$$\begin{aligned}[-\widehat{S}^m + (\widehat{S} - \rho 2^{-\delta})^m]2^\delta \leq w[i-1] = (x - \widehat{S}^m)2^\delta \\ \leq [-\widehat{S}^m + (\widehat{S} + \rho 2^{-\delta})^m]2^\delta\end{aligned}$$

and, then, $(\widehat{S} - \rho 2^{-\delta})^m \leq x \leq (\widehat{S} + \rho 2^{-\delta})^m$, that is,

$$\widehat{S} - \rho 2^{-\delta} \leq (x)^{1/m} \leq \widehat{S} + \rho 2^{-\delta}. \tag{32}$$

Since x is normalized to $2^{-m} \leq x < 1$, it is

$$1/2 \leq (x)^{1/m} \leq 1 - 2^{-n}. \tag{33}$$

In order to have convergence, (32) must guarantee (33), that is, $\widehat{S} - \rho 2^{-\delta} \geq 1/2$ and $\widehat{S} + \rho 2^{-\delta} \leq 1 - 2^{-n}$. Since $\widehat{S}$ has granularity $2^{-\delta}$, we get

$$\lfloor 1/2 + \rho 2^{-\delta} \rfloor \leq \widehat{S} \leq \lceil 1 - \rho 2^{-\delta} - 2^{-n} \rceil.$$

□

Theorem 5. *The most conservative intervals for the digit selections $s_i = k$ derived from (5), whose bounds do not depend on $i \geq \delta/b + 1$, and when the initialization is performed, providing $\widehat{S}$ on δ fractional bits, are*

$$\begin{array}{lcl} \text{for } k > 0 & \text{if} & \frac{m(k-\rho)}{r}[\min(\widehat{S} + \rho 2^{-\delta}, 1)]^{m-1} \leq \\ & w[i-1] & \leq \frac{m(k+\rho)}{r}[\max(\widehat{S} - \rho 2^{-\delta}, 2^{-1})]^{m-1} \\ \text{for } k = 0 & \text{if} & -\frac{m\rho}{r}[\max(\widehat{S} - \rho 2^{-\delta}, 2^{-1})]^{m-1} \leq \\ & w[i-1] & \leq \frac{m\rho}{r}[\max(\widehat{S} - \rho 2^{-\delta}, 2^{-1})]^{m-1} \\ \text{for } k < 0 & \text{if} & \frac{m(k-\rho)}{r}[\max(\widehat{S} - \rho 2^{-\delta}, 2^{-1})]^{m-1} \leq \\ & w[i-1] & \frac{m(k+\rho)}{r}[\min(\widehat{S} + \rho 2^{-\delta}, 1)]^{m-1}. \end{array} \tag{9}$$

Proof of Theorem 5. Let us start with $1/2 < \widehat{S} < 1$ and $k > 0$ so that the max and min functions in (9) can be easily removed. For the upper bound of the interval (5), we have

$$\begin{aligned}[-S[i-1]^m + (S[i-1] + r^{-i}(k+\rho))^m]r^{i-1} \\ = \frac{m(k+\rho)}{r} S[i-1]^{m-1} \\ + r^{-i-1} \sum_{j=2}^{m} \binom{m}{j} (k+\rho)^j r^{-ij} S[i-1]^{m-j}.\end{aligned} \tag{34}$$

Now, since the rightmost term of (34) (that is, the sum times r^{-i-1}) is positive and goes to zero for $i \to \infty$, it can be neglected, still obtaining a conservative and tight bound

$$\begin{aligned}[-S[i-1]^m + (S[i-1] + r^{-i}(k+\rho))^m]r^{i-1} \\ > \frac{m(k+\rho)}{r} S[i-1]^{m-1}.\end{aligned} \tag{35}$$

While initializing $\widehat{S} \in (1/2, 1)$ on δ fractional bits, (8) holds and, hence,

$$\frac{m(k+\rho)}{r} S[i-1]^{m-1} \geq \frac{m(k+\rho)}{r} (\widehat{S} - \rho 2^{-\delta} + \rho r^{-(i-1)})^{m-1}. \tag{36}$$

Again, a conservative and tight upper bound of the digit selection interval when $s_i = k > 0$ is obtained from (36) by neglecting the positive contribution due to $\rho r^{-(i-1)}$ and, hence,

$$\frac{m(k+\rho)}{r} S[i-1]^{m-1} > \frac{m(k+\rho)}{r} (\widehat{S} - \rho 2^{-\delta})^{m-1}. \tag{37}$$

Now, let us consider the lower bound of (5) and observe that $\alpha^\theta - \beta^\theta = (\alpha - \beta)(\sum_{j=1}^{\theta} \alpha^{\theta-j}\beta^{j-1})$:

$$\begin{aligned}[-S[i-1]^m + (S[i-1] + r^{-i}(k-\rho))^m]r^{i-1} = r^{-1} \cdot \\ (k-\rho) \sum_{j=1}^{m} \big(S[i-1] + r^{-i}(k-\rho)\big)^{m-j} S[i-1]^{j-1}.\end{aligned} \tag{38}$$

Since $k \geq 1$ and $\rho \leq 1$, it follows that (38) is not decrescent in $S[i-1]$ and, hence, the most conservative lower bound for the digit selection interval is obtained in correspondence to the upper bound of (8), that is, $S[i-1] = \widehat{S} + \rho(2^{-\delta} - r^{-(i-1)})$. Therefore,

$$\begin{aligned}[-S[i-1]^m + (S[i-1] + r^{-i}(k-\rho))^m]r^{i-1} \\ \leq (k-\rho)r^{-1} \sum_{j=1}^{m} (\widehat{S} + \rho 2^{-\delta} - \rho r^{-(i-1)})^{m-j} \cdot \\ [\widehat{S} + \rho 2^{-\delta} + r^{-i}(k - \rho - \rho r)]^{j-1}.\end{aligned}$$

We observe that $(\widehat{S} + \rho 2^{-\delta} - \rho r^{-(i-1)}) < \widehat{S} + \rho 2^{-\delta}$ and, since $1 \leq k \leq \rho(r-1)$, it is also

$$\widehat{S} + \rho 2^{-\delta} + r^{-i}(k - \rho - \rho r) < \widehat{S} + \rho 2^{-\delta}.$$

For this reason, we get

$$\begin{aligned}(k-\rho)r^{-1} \sum_{j=1}^{m} (\widehat{S} + \rho 2^{-\delta} - \rho r^{-(i-1)})^{m-j} \cdot \\ [\widehat{S} + 2^{-\delta} + r^{-i}(k - \rho - \rho r)]^{j-1} \\ < (k-\rho)r^{-1} \sum_{j=1}^{m} (\widehat{S} + \rho 2^{-\delta})^{m-j} [\widehat{S} + \rho 2^{-\delta}]^{j-1} \\ = m(k-\rho)r^{-1}(\widehat{S} + \rho 2^{-\delta})^{m-1},\end{aligned}$$

which is actually the tightest conservative lower bound of the digit selection interval when $s_i = k > 0$. The analysis of $1/2 < \widehat{S} < 1$, together with $k = 0$ and $k < 0$, produces the results given by (9), which express the widest conditions that are still conservative with (5). The analysis of the cases $\widehat{S} = 1/2$ and $\widehat{S} = 1$ is similar and, therefore, is not considered in detail here. □

Theorem 6. *The tight domain of the existence of $w[i-1]$ that is derived from the region of convergence (4), whose bounds do not depend on $i \geq \delta/b + 1$, is defined as $-L_W < w[i-1] < U_W$, where*

$$\begin{aligned} &for\ \rho<1 \qquad -\rho m < w[i-1] \le (-1+(1+\rho 2^{-\delta})^m)2^{\delta} \\ &for\ \rho=1 \qquad -m < w[i-1] < m. \end{aligned} \tag{10}$$

Proof of Theorem 6. Let us first consider the upper bound of the interval (4). We have to split our analysis into two cases: $\rho=1$ and $\rho<1$. We start with $\rho=1$ and observe that

$$\begin{aligned} &[-S[i-1]^m+(S[i-1]+r^{-(i-1)})^m]r^{i-1} \\ &\quad = r^{i-1}[-S[i-1]+S[i-1]+r^{-(i-1)}]\cdot \\ &\qquad \cdot\sum_{j=1}^{m}(S[i-1]+r^{i-1})^{m-j}S[i-1]^{j-1} \\ &\quad = \sum_{j=1}^{m}(S[i-1]+r^{i-1})^{m-j}S[i-1]^{j-1} \end{aligned} \tag{39}$$

achieves the largest value for the largest value of $S[i-1]$. We remember from (7) that $\widehat{S}\le 1-2^{-\delta}$ and, from (8), that $S[i-1]\le \widehat{S}+2^{-\delta}-r^{-(i-1)}$. Therefore, by substituting $S[i-1]=1-r^{-(i-1)}$ in (39) and by observing that $(1-r^{i-1})^j<1$, we get the tight conservative condition (40) for the upper bound of the domain of $w[i-1]$:

$$\sum_{j=1}^{m}[(S[i-1]+r^{i-1})^{m-j}S[i-1]^{j-1}] \le \sum_{j=1}^{m}(1-r^{i-1})^j < m. \tag{40}$$

For $\rho<1$, the approach is similar and we observe that the upper bound of (4) achieves the largest value for the largest value of $S[i-1]$. Since $\widehat{S}\le 1$ from (7) and $S[i-1]\le \min(1,\widehat{S}+\rho(2^{-\delta}-r^{-(i-1)})]$ from (8), we substitute $S[i-1]=1$ in the upper bound of (4):

$$\begin{aligned} &[-S[i-1]^m+(S[i-1]+\rho r^{-(i-1)})^m]r^{i-1} \\ &\qquad \le (-1+(1+\rho r^{-(i-1)})^m)r^{i-1} \\ &\quad = r^{i-1}(-1+1+\rho r^{-(i-1)})\sum_{j=1}^{m}\left(1+\rho r^{-(i-1)}\right)^{j-1} \\ &\qquad = \rho\sum_{j=1}^{m}\left(1+\rho r^{-(i-1)}\right)^{j-1}, \end{aligned} \tag{41}$$

from which it follows that $r^{-(i-1)}$ must be made as large as possible. Since, from $i=\delta b+p$, we have $r^{-(i-1)}\le 2^{-\delta}$, its substitution in (41) leads to the tight conservative upper bound:

$$(-1+(1+\rho r^{-(i-1)})^m)r^{i-1} \le (-1+(1+\rho 2^{-\delta})^m)2^{\delta}.$$

For the lower bound, it is not necessary to split the analysis in the cases $\rho=1$ and $\rho<1$:

$$\begin{aligned} &[-S[i-1]^m+(S[i-1]-\rho r^{-(i-1)})^m]r^{i-1} \\ &\quad = r^{i-1}[-S[i-1]+S[i-1]-\rho r^{-(i-1)}] \\ &\qquad \cdot\sum_{j=1}^{m}(S[i-1]-\rho r^{-(i-1)})^{m-j}S[i-1]^{j-1} \\ &\quad = -\rho\sum_{j=1}^{m}(S[i-1]-\rho r^{-(i-1)})^{m-j}S[i-1]^{j-1}. \end{aligned} \tag{42}$$

We observe that (42) achieves the smallest value for the largest value of $S[i-1]$. Therefore, from (8), by substituting $S[i-1]=1$, we get the tight conservative lower bound,

$$\begin{aligned} &-\rho\sum_{j=1}^{m}(S[i-1]-\rho r^{-(i-1)})^{m-j}S[i-1]^{j-1} \\ &\qquad \ge -\rho\sum_{j=1}^{m}(1-\rho r^{-(i-1)})^{m-j} > -\rho m, \end{aligned} \tag{43}$$

since the term in the sum is less than 1 and approaches 1 for $i\to\infty$. □

Theorem 7. *In order to have correct digit selection rules, in the case of the carry-save representation of $w[i-1]$, a necessary (but not sufficient) condition is to have both an estimate $\widehat{w[i-1]}$ of $w[i-1]$ up to its tth fractional bit and consider an initial value $\widehat{S}$ on δ fractional bits given by*

$$\begin{aligned} &\frac{m(\rho r-1)}{r}(2^{-1}+(1-\rho)2^{-\delta})^{m-1} \\ &-\frac{m\rho(r-2)}{r}(2^{-1}+(1+\rho)2^{-\delta})^{m-1}-2^{-t}\ge 0, \end{aligned} \tag{15}$$

that is, with

$$\begin{aligned} t \ge \log_2\{r/\{m[&(\rho r-1)(2^{-1}+(1-\rho)2^{-\delta})^{m-1} \\ &-\rho(r-2)(2^{-1}+(1+\rho)2^{-\delta})^{m-1}]\}\}, \end{aligned} \tag{16}$$

$$\delta > \log_2\frac{2(1+\rho-(1-\rho)[(\rho r-1)/(\rho r-2\rho)]^{1/(m-1)})}{[(\rho r-1)/(\rho r-2\rho)]^{1/(m-1)}-1}. \tag{17}$$

Proof of Theorem 7. Let us first consider $1/2<\widehat{S}<1$. We assume that $k>0$. The extension to the other cases is straightforward. From (9), we have obtained (14), that is,

$$\begin{aligned} L_{[k]} &= \frac{m(k-\rho)}{r}(\widehat{S}+\rho 2^{-\delta})^{m-1}, \\ U_{[k-1]} &= \frac{m(k-1+\rho)}{r}(\widehat{S}-\rho 2^{-\delta})^{m-1}, \end{aligned}$$

while, for the carry-save representation of $w[i-1]$, we consider (13), which implies that it is necessary to have $U_{[k-1]}-2^{-t}\ge L_{[k]}$, that is,

$$\begin{aligned} &\frac{m(k-1+\rho)}{r}(\widehat{S}-\rho 2^{-\delta})^{m-1} \\ &-\frac{m(k-\rho)}{r}(\widehat{S}+\rho 2^{-\delta})^{m-1}-2^{-t}\ge 0\ . \end{aligned} \tag{44}$$

Now, (44) is a function of k and $\widehat{S}$. By assuming having a fixed $\widehat{S}$, the left-hand side of (44) decreases as $k>0$ increases. Therefore, the worst case in (44) occurs for $k=a=\rho(r-1)$. Let us study the function obtained from the left-hand side of (44), without the term 2^{-t}, that is,

$$F(\widehat{S}) = \frac{m(k-\rho)}{r}(\widehat{S}-\rho 2^{-\delta})^{m-1}\left[\frac{(k-1+\rho)}{(k-\rho)} - \left(1+\frac{\rho 2^{-\delta+1}}{\widehat{S}-\rho 2^{-\delta}}\right)^{m-1}\right]. \quad (45)$$

This function expresses the overlap [7] between the two consecutive digit selections $k-1$ and k. Now, let us study (45) by assuming that, since we are looking for the most critical value of $\widehat{S}$, k is kept fixed. We know that $F(\widehat{S})$ must be positive for all of the domains of $\widehat{S}$ in order to have a valid overlap and, hence, digit selections. We observe that $F(\widehat{S})$ is crescent in $\widehat{S}$ since both $(\widehat{S}-\rho 2^{-\delta})^{m-1}$ and

$$\left[\frac{(k-1+\rho)}{(k-\rho)} - \left(1+\frac{\rho 2^{-\delta+1}}{\widehat{S}-\rho 2^{-\delta}}\right)^{m-1}\right]$$

are crescent in $\widehat{S}$. This implies that the worst case in (45) and (44) occurs for the smallest $\widehat{S}$. By substituting, in (44), $k=a=\rho(r-1)$ and $\widehat{S}=1/2+2^{-\delta}$, we get

$$\frac{m(\rho r-1)}{r}(2^{-1}+(1-\rho)2^{-\delta})^{m-1} - \frac{m\rho(r-2)}{r}(2^{-1}+(1+\rho)2^{-\delta})^{m-1} - 2^{-t} \geq 0$$

(that is, (15)), which expresses the condition on the overlap to be satisfied in order to have valid digit selection rules. From (15), we get the following necessary condition (16) on t:

$$t \geq \log_2\{r/\{m[(\rho r-1)(2^{-1}+(1-\rho)2^{-\delta})^{m-1} - \rho(r-2)(2^{-1}+(1+\rho)2^{-\delta})^{m-1}]\}\}.$$

From (16), we observe that it must be (for $r>2$)

$$[(\rho r-1)(2^{-1}+(1-\rho)2^{-\delta})^{m-1} - \rho(r-2)(2^{-1}+(1+\rho)2^{-\delta})^{m-1}] \geq 0,$$

that is (17),

$$\delta > \log_2 \frac{2(1+\rho-(1-\rho)[(\rho r-1)/(\rho r-2\rho)]^{1/(m-1)})}{[(\rho r-1)/(\rho r-2\rho)]^{1/(m-1)}-1}.$$

Therefore, in the case of the carry-save representation of $w[i-1]$, it is necessary for t and δ to satisfy (16) and (17), respectively. Observe that (17) is valid for $r>2$. For $r=2$, it was shown in Section 5 (see Theorems 10 and 11) that no lookup table for $\widehat{S}$ is necessary. The analysis of the other cases $k=0$ and $k<0$ provides results that are conservative with (16) and (17).

The study of the case $\widehat{S}=1/2$ and $k>0$ yields the necessary condition:

$$\frac{m(\rho r-1)}{r}(2^{-1})^{m-1} - \frac{m\rho(r-2)}{r}(2^{-1}+\rho 2^{-\delta})^{m-1} - 2^{-t} \geq 0\,. \quad (46)$$

Now, the point is to show that (46) is less restrictive than (15), that is, that the left-hand side of (46) is larger than the corresponding left-hand side of (15). To do this, we subtract the left-hand side of (15) from the left-hand side of (46) and show that the result is greater than or equal to zero. After some simple manipulations, we get

$$(2^{-1}+(1+\rho)2^{-\delta})^{m-1} - (2^{-1}+\rho 2^{-\delta})^{m-1} \geq \frac{\rho r-1}{\rho(r-2)}\left[(2^{-1}+(1-\rho)2^{-\delta})^{m-1} - (2^{-1})^{m-1}\right].$$

Then, by observing that

$$\alpha^\theta - \beta^\theta = (\alpha-\beta)\left(\sum_{j=1}^{\theta}\alpha^{\theta-j}\beta^{j-1}\right),$$

we get

$$2^{-\delta}\sum_{j=1}^{m-1}(2^{-1}+(1+\rho)2^{-\delta})^{m-1-j}(2^{-1}+\rho 2^{-\delta})^{j-1} \geq \frac{\rho r-1}{\rho(r-2)}(1-\rho)2^{-\delta}\sum_{j=1}^{m-1}(2^{-1}+(1-\rho)2^{-\delta})^{m-1-j}(2^{-1})^{j-1},$$

where we observe that the left-hand side sum is greater than the right-hand side sum since each term is greater in the first sum than the corresponding in the second sum. In addition, we observe that $2^{-\delta} \geq \frac{\rho r-1}{\rho(r-2)}(1-\rho)2^{-\delta}$.[3] This confirms that (15) is more restrictive than (46). Therefore, (15), (16), and (17) are still valid. The analysis of the other cases $k \leq 0$ and $\widehat{S}=1$ provides results that are conservative with (15), (16), and (17). □

Theorem 8. *The lookup table intervals are*

$$\textit{lookup } \widehat{S}=v \textit{ if } (v-\rho 2^{-\delta})^m \leq x \leq (v+\rho 2^{-\delta})^m. \quad (20)$$

Proof of Theorem 8. The starting point is the expression of the region of convergence (4). By substituting (3) in (4), we get

$$(S[i-1]-\rho r^{-(i-1)})^m \leq x \leq (S[i-1]+\rho r^{-(i-1)})^m. \quad (47)$$

Now, the lookup process which occurs for $i=\delta/b+1$ must ensure that the estimate $\widehat{S}$ is selected such that (47) holds. By calling v the generic value for $\widehat{S}$, and by substituting in (47) (as well as $r^{-(i-1)}=2^{-\delta}$), we get the lookup table intervals expressed by (20). □

Theorem 9. *In order to have correct lookup table rules, a sufficient (but not necessary) condition is to have an estimate $\widehat{x}$ of x up to its uth fractional bit, given by*

$$u \geq -\log_2[(2^{-1}-(1-\rho)2^{-\delta})^m - (2^{-1}-\rho 2^{-\delta})^m]. \quad (23)$$

Proof of Theorem 9. Let us apply (22) for the two consecutive values of $\widehat{S}$, v and $v+2^{-\delta}$. We get $(v-(1-\rho)2^{-\delta})^m - (v-\rho 2^{-\delta})^m \geq 2^{-u}$, which can be easily observed to be crescent in v. Therefore, the most critical case occurs for $v=2^{-1}$, which leads to (23). □

Theorem 10. *When $r=2$, it is possible to start the iterations for the mth root extraction process by considering $i=3$, $\widehat{S}=S[2]=2^{-1}+2^{-2}$, and $w[2]=(x-S[2]^m)2^2$.*

3. In fact, after some passages, it leads to $\rho^2 r-3\rho+1 \geq 0$, which is certainly true for $r>2$, as assumed here.

Proof of Theorem 10. We prove that, with $r = 2$, if $S[2] = 3/4$, then $w[2] = (x - S[2]^m) \cdot 2^2$ certainly belongs to the region of convergence, which is derived in such a case given by (4), that is,

$$[-(3/4)^m + (3/4 - 2^{-2})^m]2^2 = [-(3/4)^m + (1/2)^m]2^2 \leq w[2] \leq [-(3/4)^m + (3/4 + 2^{-2})^m]2^2 = [1 - (3/4)^m]2^2. \quad (48)$$

When x is at the lower bound of its domain (that is, $x = 2^{-m}$), we have $w[2] = [-(3/4)^m + (1/2)^m]2^2$, which corresponds to the lower bound of (48). In addition, when $x = 1$, it is at the upper bound of its domain and we have $w[2] = [1 - (3/4)^m]2^2$, which is equal to the upper bound of (48). Since $w[2]$ is monotonic in x, it follows that, for all of the x values belonging to the interval $2^{-m} \leq x < 1$, $w[2]$ belongs to the region of convergence (48). □

Theorem 11. *For $r = 2$, the digit selection intervals are*

$$\begin{array}{rll} s_i = 1 & if & 0 \leq w[i-1] < m \\ s_i = 0 & if & -m2^{-m} \leq w[i-1] \leq m2^{-m} \\ s_i = -1 & if & -m < w[i-1] \leq 0. \end{array} \quad (24)$$

Proof of Theorem 11. Let us consider the expression of the digit selection intervals given by (9) and, for the bounds of the region of convergence, the results of Theorem 6. We observe that, thanks to Theorem 10, when $r = 2$, the value $\widehat{S}$ in (9) is substituted by $3/4$ (with $\delta = 2$) and the expressions in (24) are consequently obtained. Observe that the selection intervals of (24) (and, hence, also the rules that will be derived from this) do not depend on δ and on $\widehat{S}$. □

Theorem 12. *For $r = 2$, it is necessary and sufficient to use the estimate $w[\widehat{i-1}]$ obtained by considering $w[i-1]$ up to its $t = m - \lfloor \log_2(m) \rfloor$ fractional bit in order to obtain valid digit selection rules, which, for a carry-save representation of the residual $w[\widehat{i-1}]$, are*

$$\begin{array}{rll} s_i = +1 & if & 0 \leq w[\widehat{i-1}] \leq m - 2^{-m+f} \\ s_i = 0 & if & w[\widehat{i-1}] = -2^{-m+f} \\ s_i = -1 & if & -m - 2^{-m+f} \leq w[\widehat{i-1}] \leq -2^{-m+f+1}, \end{array}$$

with $f = \lfloor \log_2 m \rfloor$.

Proof of Theorem 12. As in Section 3.6, we use $L_{[k]}$ and $U_{[k]}$, which define the lower and upper bounds of the domain of $w[i-1]$ related to the generic digit selection $s_i = k$ given by (24) to determine the set of constants $g_{[k]}$. By replacing the values of the bounds for the case $k = 1$, we get

$$L_{[1]} = 0 \leq g_{[1]} \leq m \cdot 2^{-m} - 2^{-t} = U_{[0]} - 2^{-t}. \quad (49)$$

A necessary and sufficient condition is to have $L_{[k]} \leq U_{[k-1]} - 2^{-t}$ in (13) (and, hence, in (49) too). Consequently, the minimum value for t that produces a valid set of digit selection rules is

$$t = \lceil m - \log_2(m) \rceil = m - \lfloor \log_2(m) \rfloor = m - f \geq 1, \quad (50)$$

where $f = \lfloor \log_2 m \rfloor$. With this value of t, from (49), we can use $g_{[1]} = 0$. By now considering the case $k = 0$, we get $L_{[0]} = -m \cdot 2^{-m} \leq g_{[0]} \leq 0 - 2^{-t} = U_{[-1]} - 2^{-t}$, which again leads to (50) and to $g_{[0]} = -2^{-t}$. The domain of $w[i-1]$ is provided by Theorem 6 and (10) for $\rho = 1$. Therefore, the domain of $w[\widehat{i-1}]$ is identified in

$$-m - 2^{-t} \leq w[\widehat{i-1}] \leq m - 2^{-t}. \quad (51)$$

All of the above leads to the digit selection rules given by (25). □

Theorem 13. *The estimate $w[\widehat{i-1}]$ is obtained by the most significant $m + 2$ bits of $w[i-1]$.*

Proof of Theorem 13. Let us denote with c the number of integer bits of $w[\widehat{i-1}]$ and with g the global number of bits required to represent $w[\widehat{i-1}]$. From (25), (50), and (51), by observing that the lower bound is the most "critical," we have $c = 1 + \lceil \log_2(m + 2^{-m+f}) \rceil$. We know from (50) that the term $2^{-m+f} < 1$ for $m \geq 1$. Therefore, by observing that, from the definition of f, we have $2^f \leq m < 2^{f+1}$, we get $c = 1 + \lceil \log_2(m + 2^{-m+f}) \rceil = f + 2$ integer bits, which then gives the following value as the global number of bits:

$$g = c + t = f + 2 + m - f = m + 2. \quad (52)$$

This implies that the number of bits of $w[i-1]$ to be considered for the estimate increases linearly with the value of m. □

ACKNOWLEDGMENTS

The authors would like to thank Elisardo Antelo, Alberto Nannarelli, Piera, Lia and Gaia, Mario and Mariele, Ale Boga, Derna and Marcello, and Belén Espiña for their support. This work was done while J.-A. Piñeiro was with the University of Santiago de Compostela. The work of J.-A. Piñeiro and J.D. Bruguera was financially supported by the Ministry of Science and Technology (MCYT) Contract TIC2001-3694-C02.

REFERENCES

[1] E. Antelo, T. Lang, and J. Bruguera, "Computation of $\sqrt{x/d}$ in a Very High Radix Combined Division/Square-Root Unit with Scaling," *IEEE Trans. Computers,* vol. 47, no. 2, pp. 152-161, Feb. 1998.

[2] L. Ciminiera and P. Montuschi, "Higher Radix Square Rooting," *IEEE Trans. Computers,* vol. 39, no. 10, pp. 1220-1231, Oct. 1990.

[3] J. Cortadella and T. Lang, "High-Radix Division and Square Root with Speculation," *IEEE Trans. Computers,* vol. 43, no. 8, pp. 919-931, Aug. 1994.

[4] D. DasSarma and D.W. Matula, "Faithful Bipartite ROM Reciprocal Tables," *Proc. 12th IEEE Symp. Computer Arithmetic,* pp. 12-25, July 1995.

[5] M.D. Ercegovac, "A General Hardware-Oriented Method for Evaluation of Functions and Computations in a Digital Computer," *IEEE Trans. Computers,* vol. 26, no. 7, pp. 667-680, July 1977.

[6] M.D. Ercegovac and T. Lang, "On-the-Fly Conversion of Redundant into Conventional Representations," *IEEE Trans. Computers,* vol. 36, no. 7, pp. 895-897, July 1987.

[7] M.D. Ercegovac and T. Lang, "Radix-4 Square Root without Initial PLA," *IEEE Trans. Computers,* vol. 39, no. 8, pp. 1016-1024, Aug. 1990.

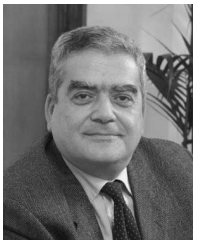
[8] M.D. Ercegovac and T. Lang, *Algorithms for Division and Square Root.* Kluwer Academic, 1994.

[9] M.D. Ercegovac and T. Lang, *Digital Arithmetic.* Morgan Kaufmann, 2003.

[10] A.S. Glassner, *Principles of Digital Image Synthesis.* Morgan Kaufmann, 1995.
[11] J.-M. Muller, *Elementary Functions Algorithms and Implementation.* Birkhauser, 1997.
[12] S.F. Oberman, "Floating Point Division and Square Root Algorithms and Implementation in the AMD-K7 Microprocessor," *Proc. 14th IEEE Symp. Computer Arithmetic,* pp. 106-115, July 1999.
[13] J.-A. Piñeiro, M.D. Ercegovac, and J.D. Bruguera, "Algorithm and Architecture for Logarithm, Exponential, and Powering Computation," *IEEE Trans. Computers,* vol. 53, no. 9, pp. 1085-1096, Sept. 2004.
[14] J.-A. Piñeiro, J.D. Bruguera, L. Ciminiera, and P. Montuschi, "A Digit-by-Digit Algorithm for Radix-2 Cube Root and Its Implementation," technical report, http://www.ac.usc.es, 2004.
[15] M.J. Schulte and J.E. Stine, "The Symmetric Table Addition Method for Accurate Function Approximation," *J. VLSI Signal Processing,* vol. 21, no. 2, pp. 167-177, 1999.
[16] M.J. Schulte and J.E. Stine, "Approximating Elementary Functions with Symmetric Bipartite Tables," *IEEE Trans. Computers,* vol. 48, no. 8, pp. 842-847, Aug. 1999.
[17] N. Takagi, "A Digit-Recurrence Algorithm for Cube Rooting," *IEICE Trans. Fundamentals,* vol. E84-A, no. 5, pp. 1309-1314, May 2001.

Paolo Montuschi received the bachelor's degree in electronic engineering and the PhD degree in computer engineering from the Politecnico di Torino, Italy, in 1984 and 1989, respectively. From 1988 to 1994, he was a member of the Board of the Italian Association for Computer Graphics. From 1995 to 1997 and in 2006, he was the deputy chairman of the Center for Computing Facilities and Services at the Politecnico di Torino. Since January 2000, he has been a full professor at the Politecnico di Torino and, since 2003, he has been the chairman of the Control and Computer Engineering Department at the Politecnico di Torino. He served on the program committees of the 13th through 18th IEEE Symposium on Computer Arithmetic and was a program cochair of the 2005 symposium. From 2000 to 2004, he served as an associate editor of the *IEEE Transactions on Computers*. His current research interests include computer arithmetic, with a special emphasis on algorithms and architectures for fast elementary function evaluations, and computer graphics. He is a senior member of the IEEE and a member of the IEEE Computer Society.

Javier D. Bruguera received the BS degree in physics and the PhD degree from the University of Santiago de Compostela, Spain, in 1984 and 1989, respectively. He is a professor in the Department of Electronic and Computer Science at the University of Santiago de Compostela. Previously, he was an assistant professor in the Department of Electrical, Electronic, and Computer Engineering at the University of Oviedo, Spain, and an assistant professor in the Department of Electronic Engineering at the University of La Coruña, Spain. He was a visiting researcher in the Application Center of Microelectronics at Siemens, Munich, Germany, and in the Department of Electrical Engineering and Computer Science at the University of California, Irvine. His primary research interests are computer arithmetic, processor design, digital design for signal and image processing, and computer architecture. He is a member of the IEEE and the IEEE Computer Society.



Luigi Ciminiera received the degree in electronic engineering from the Politecnico di Torino, Italy, in 1977. From 1978 to 1983, he held different positions with the Department of Control and Computer Engineering at the Politecnico di Torino, where he became a research assistant in 1983. He was the chairman of the Control and Computer Engineering Department at the Politecnico di Torino from 1999 to 2003 and he is the dean of the II School of Engineering at the Politecnico di Torino. He served as a program cochair of the 14th IEEE Symposium on Computer Arithmetic. His research interests include grids and peer-to-peer networks, distributed software systems, and computer arithmetic. He is a coauthor of two international books and more than 100 contributions published in technical journals and conference proceedings. He is a member of the IEEE.

José-Alejandro Piñeiro received the BSc degree, the MSc degree in physics (electronics), and the PhD degree in computer engineering from the University of Santiago de Compostela, Spain, in 1998, 1999, and 2003, respectively. Since 2004, he has been with the Intel Barcelona Research Center (IBRC), Intel Labs-UPC. His research interests are computer arithmetic, memory subsystems, computer graphics, multimedia, game physics, and numerical processors.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.