Digit Serial Methods with Applications to Division and Square Root

Warren E. Ferguson Jr., Jesse Bingham, Levent Erkök, John R. Harrison, and Joe Leslie-Hurd

Abstract—We present a generic digit serial method (DSM) to compute the digits of a real number V. Bounds on these digits, and on the errors in the associated estimates of V formed from these digits, are derived. To illustrate our results, we derive such bounds for a parameterized family of high-radix algorithms for division and square root. These bounds enable a DSM designer to determine, for example, whether a given choice of parameters allows rapid formation and rounding of its approximation to V.

Index Terms—Digit serial method, digit recurrence method, mixed-radix representation, on-the-fly technique, high-radix, division, square root, digit bounds, error bounds

1 Introduction

Let V be a real number. A digit serial method (DSM) is an algorithm that determines the digits of V serially, starting with the leading digit. A DSM begins by initializing an accumulator to zero and, as each digit is determined, that digit is aligned and added to the accumulator. Successive values of this accumulator form a sequence of estimates of V.

The primary contribution of this paper is a generic DSM analysis method for determining bounds on the magnitudes of the digits, as well as bounds on the error associated with the estimates of V. These bounds allow a designer to determine the required bit-width of signals representing these digits and errors, and to determine when the estimates of V can be efficiently formed and rounded by, say, on-the-fly techniques.

The major results presented here are the Proxy Theorem 5.1 and its Corollary 5.3 with illustrations of their application to division and square root algorithms. These results have been checked/formalized using the HOL Light [1] theorem prover; a short extract from the formalization is presented in Section 10.

The analysis of low-radix DSM for division and square root is well-understood [2]. Analyses of specific high-radix DSM for these operations are described in [3], [4], [5]. An additional contribution of this paper is the application of our generic DSM analysis to a parameterized family of high-radix DSM algorithms for division and square root.

2 Scaling

The DSM considered here assume that $V \in (0,1)$, so the leading digit of V is known to be the first fraction digit. For this assumption to be true, it may be necessary to scale the problem. Scaling is a three step process: (1) reduce the general problem to simpler problem by scaling, (2) determine the result of the simpler problem, and (3) reconstruct the general result from the result of the simpler problem.

For completeness, we briefly describe well-known scalings for division and square root of positive normalized finite precision binary floating-point numbers. Here, a positive normalized finite precision binary floating-point number is a real value of the form $s2^e$ composed of a normalized significand $s = 1 + f/2^k$, an integer exponent e, and a fraction $f/2^k$ where f is a non-negative integer less than 2^k for some positive integer k.

Scaling for division. Consider the computation of the quotient $Q \equiv (s_x 2^{e_x})/(s_y 2^{e_y})$ where s_x and s_y are normalized finite precision binary significands, and e_x and e_y are integers. Scaling reduces the computation of Q to the computation of a related quotient $V \in (0,1)$, a DSM is used to compute V, and Q is reconstructed from the value of V. One possible scaling uses the reduction

$$V \equiv X/Y \quad \text{where} \quad (X,Y) \equiv (s_x/2, s_y),$$

so $X \in [1/2,1)$, $Y \in [1,2)$, and $V \in (1/4,1)$. After the DSM determines V, the final result is reconstructed as follows:

$$Q = V 2^{e_x - e_y + 1}.$$

Scaling for square-root. Consider the computation of the square root $R \equiv \sqrt{s_x 2^{e_x}}$ where s_x is a normalized finite precision binary significand and e_x is an integer. Scaling reduces the computation of R to the computation of a related square root $V \in (0,1)$, a DSM is used to compute V, and R is reconstructed from the value of V. One possible scaling uses the reduction

$$V \equiv \sqrt{X} \quad \text{where} \quad X \equiv \begin{cases} s_x/4 & \text{even } e_x \\ s_x/2 & \text{odd } e_x, \end{cases}$$

so $X \in [1/4,1)$ and $V \in [1/2,1)$. After the DSM determines V, the final result is reconstructed as follows:

$$R = V \begin{cases} 2^{(e_x+2)/2} & \text{even } e_x \\ 2^{(e_x+1)/2} & \text{odd } e_x. \end{cases}$$

For both division and square root, scaling has reduced the original problem to the computation of a value $V \in (0,1)$, combined with integer additions that determine the associated exponent.

3 Basic DSM

Consider the following mixed-radix representation of a real number V

$$\begin{aligned} V &= \frac{1}{\beta_1}\left(v_1 + \frac{1}{\beta_2}\left(v_2 + \frac{1}{\beta_3}\left(v_3 + \cdots\right)\right)\right) \\ &= \frac{v_1}{B_1} + \frac{v_2}{B_2} + \frac{v_3}{B_3} + \cdots, \end{aligned}$$

where[1] $\forall i \in \mathbb{N}^{>0} : B_i \equiv \beta_1 \beta_2 \ldots \beta_i$. We always assume that $\{v_i\}_{i=1}^{\infty}$ is a sequence of integers (called *digits*), and that $\{\beta_i\}_{i=1}^{\infty}$ is a sequence of integers (called *radices* or *bases*), each 2 or greater. If $B_0 \equiv 1$, then $\forall i \in \mathbb{N} : B_{i+1} = \beta_{i+1} B_i$. As illustrated in Section 8 for square root, a DSM utilizing a mixed-radix, rather than fixed-radix, representation offers additional control over the upper bounds on the magnitudes of the digits it produces. Ercegovac and Muller [6] have also demonstrated the benefits of using a mixed-radix representation for real and complex division.

- W. Ferguson is retired from Intel Corporation, Hillsboro, OR 97124. E-mail: warren_e_ferguson@hotmail.com.
- J. Bingham, L. Erkök, J.R. Harrison, and J. Leslie-Hurd are with Intel Corporation, Hillsboro, OR 91724. E-mail: {jesse.bingham, erkokl}@gmail.com, johnh@ichips.intel.com, joe@leslie-hurd.com.

Manuscript received 3 Aug. 2017; revised 29 Sept. 2017; accepted 2 Oct. 2017. Date of publication 5 Oct. 2017; date of current version 16 Feb. 2018.
(Corresponding author: Warren E. Ferguson.)
Recommended for acceptance by J.-M. Muller.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2017.2759764

[1] Notation: Reals $\mathbb{R}$, non-negative reals $\mathbb{R}^{\geq 0}$, positive reals $\mathbb{R}^{>0}$, integers $\mathbb{Z}$, natural numbers $\mathbb{N} = \{0, 1, \ldots\}$, counting numbers $\mathbb{N}^{>0} = \{1, 2, \ldots\}$.

A DSM accumulates the terms of the series for V serially. Start with an accumulator initialized to 0. The terms involving the digits $v_1, v_2, v_3, \ldots$ are then consecutively added to the accumulator. The values of the accumulator after each digit is added defines the *head* sequence $\{H_i\}_{i=0}^{\infty}$ where

$$H_0 \equiv 0,\ \forall i \in \mathbb{N}^{>0} : H_i \equiv \frac{v_1}{B_1} + \frac{v_2}{B_2} + \cdots + \frac{v_i}{B_i}.$$

Associated with each head H_i is the *tail* T_i defined as

$$\forall i \in \mathbb{N} : T_i \equiv B_i(V - H_i) = B_i\left(\frac{v_{i+1}}{B_{i+1}} + \frac{v_{i+2}}{B_{i+2}} + \cdots\right).$$

Intuitively, H_i is the approximation to the target result V that has been computed after step i, while T_i is the error in this approximation normalized by B_i; here T_i/B_i is analogous to a floating-point value $s2^e$ with $T_i \sim s$ and $1/B_i \sim 2^e$. This definition of the tails provides the invariant $\forall i \in \mathbb{N} : V = H_i + T_i/B_i$.

We can summarize the above as follows:

$$\begin{aligned}
&B_0 = 1,\ \forall i \in \mathbb{N} : B_{i+1} = \beta_{i+1}B_i,\\
&H_0 = 0,\ \forall i \in \mathbb{N} : H_{i+1} = H_i + v_{i+1}/B_{i+1},\ \text{and}\\
&T_0 = V,\ \forall i \in \mathbb{N} : T_{i+1} = \beta_{i+1}T_i - v_{i+1}.
\end{aligned}$$

Digit selection. In the recurrence

$$\forall i \in \mathbb{N} : \beta_{i+1}T_i = v_{i+1} + T_{i+1},$$

note that

$$T_{i+1} = \frac{v_{i+2}}{\beta_{i+2}} + \frac{v_{i+3}}{\beta_{i+2}\beta_{i+3}} + \cdots.$$

As we shall see in Section 4, if the digits satisfy $\forall k \geq 2 : |v_k| < \beta_k$, a simple algorithm can be used to accumulate the digits. If this condition holds then $|T_{i+1}|$, the distance between $\beta_{i+1}T_i$ and v_{i+1}, is at most 1. Consequently, a plausible choice for v_{i+1} is an integer near $\beta_{i+1}T_i$.

We therefore introduce *digit selection functions* $\forall i \in \mathbb{N}^{>0} : \mathrm{DSF}_i : \mathbb{R} \to \mathbb{Z}$ that "round" their real argument to a nearby integer, so $\forall i \in \mathbb{N} : v_{i+1} \equiv \mathrm{DSF}_{i+1}(\beta_{i+1}T_i)$. Paired with any digit selection function DSF is the *complementary digit selection function* $\mathrm{coDSF} : \mathbb{R} \to \mathbb{R}$ defined as

$$\forall z \in \mathbb{R} : \mathrm{coDSF}(z) \equiv z - \mathrm{DSF}(z).$$

From the partition

$$\beta_{i+1}T_i = \mathrm{DSF}_{i+1}(\beta_{i+1}T_i) + \mathrm{coDSF}_{i+1}(\beta_{i+1}T_i),$$

of $\beta_{i+1}T_i$, we recognize that $T_{i+1} = \mathrm{coDSF}_{i+1}(\beta_{i+1}T_i)$.

Note that $|\mathrm{coDSF}(z)|$ is the distance between z and $\mathrm{DSF}(z)$, or equivalently the error in approximating z by $\mathrm{DSF}(z)$. It makes sense, then, to classify digit selection functions by the maximum value of $|\mathrm{coDSF}(z)|$ for all z.

Definition 3.1 (Round to Nearby Integer). *For* $\Omega \in \mathbb{R}$, $\mathrm{RNI}(\Omega)$ *is the collection of all digit selection functions* $\mathrm{DSF} : \mathbb{R} \to \mathbb{Z}$ *such that* $\forall z \in \mathbb{R} : |\mathrm{coDSF}(z)| \leq \Omega$.

Observe that the round-to-nearest integer function is an element of $\mathrm{RNI}(\Omega)$ whenever $\Omega \geq 1/2$. Suppose $\mathrm{DSF} \in \mathrm{RNI}(\Omega)$, so $\forall z \in \mathbb{R} : \mathrm{DSF}(z) \in [z - \Omega, z + \Omega] \cap \mathbb{Z}$. The closed interval $[z - \Omega, z + \Omega]$ always contains at least one integer because its length 2Ω is at least 1. For example, when $1/2 \leq \Omega < 1$, this interval contains one and sometimes two integers.

The ideas underlying exclusion zones [7] for division and square root suggest how the definition of $\mathrm{RNI}(\Omega)$ might be modified so it is nonempty for $\Omega < 1/2$. Suppose, for example, that a digit selection function's argument is never closer than a distance ϵ to a half-integer (half an odd integer). Let $\mathbb{F}$ be $\mathbb{R}$ after open intervals of radius ϵ centered at each half-integer have been removed (excluded). Then $\mathrm{RNI}(\Omega)$ should be defined as the collection of all digit selection functions $\mathrm{DSF} : \mathbb{F} \to \mathbb{Z}$ such that $\forall z \in \mathbb{F} : |\mathrm{coDSF}(z)\ | \leq \Omega$. For example, the round-to-nearest integer function satisfies this requirement when $\Omega = 1/2 - \epsilon$.

```
procedure DSM_BASIC($V$)
  $(B_0, H_0, T_0) := (1, 0, V)$
  for $i := 0, 1, 2, \ldots$ do
    $\{$Invariant: $V = H_i + T_i/B_i\}$
    $v_{i+1} := \mathrm{DSF}_{i+1}(\beta_{i+1}T_i)$
    $B_{i+1} := \beta_{i+1}B_i$
    $H_{i+1} := H_i + v_{i+1}/B_{i+1}$
    $T_{i+1} := \beta_{i+1}T_i - v_{i+1}$
  end for
end procedure
```

Fig. 1. Basic DSM for $V \in \mathbb{R}, \forall i \in \mathbb{N}^{>0} : \beta_i \geq 2$, and $\forall i \in \mathbb{N}^{>0} : \mathrm{DSF}_i \in \mathrm{RNI}(\Omega_i)$.

Theorem 3.2. *If* $v \equiv \mathrm{DSF}(z)$ *where* $\mathrm{DSF} \in \mathrm{RNI}(\Omega)$, *then* $|v| \leq \lfloor |z| + \Omega \rfloor$.

Proof. Since $\forall x : |\mathrm{coDSF}(x)| \leq \Omega$ and $v = \mathrm{DSF}(z)$, applying the triangle inequality yields

$$|v| = |\mathrm{DSF}(z)| = |z - \mathrm{coDSF}(z)| \leq |z| + \Omega.$$

The result follows by applying the floor function to the inequality and using the fact that v is an integer. □

Procedure DSM_BASIC in Fig. 1 is the result of combining the information presented above.[2] The digit selection function can change with each iteration, all that is supposed is that $\forall i > 0 : \mathrm{DSF}_i \in \mathrm{RNI}(\Omega_i)$. For this algorithm, bounds on the absolute error $|T_i|/B_i$ in the estimate H_i of V, and on the digit v_i, are easy to derive. We know that

$$|T_0| = V \ \text{ and } \ \forall i \in \mathbb{N}^{>0} : |T_i| \leq \Omega_i,$$

because $\forall i \in \mathbb{N}^{>0} : T_i = \mathrm{coDSF}_i(\beta_i T_{i-1})$ where $\mathrm{DSF}_i \in \mathrm{RNI}(\Omega_i)$, and so applying Theorem 3.2 yields the digit bounds

$$\forall i \in \mathbb{N}^{>0} : |v_i| \leq \begin{cases} \lfloor \beta_1 V + \Omega_1 \rfloor & \text{if } i = 1\\ \lfloor \beta_i \Omega_{i-1} + \Omega_i \rfloor & \text{if } i > 1. \end{cases}$$

When the sequence $\{\Omega_i\}_{i=1}^{\infty}$ is bounded, so too is the tail sequence $\{T_i\}_{i=0}^{\infty}$. The following result proves that the head sequence converges to V if the tail sequence is bounded.

Theorem 3.3. *Let* V *and* $\{\beta_i\}_{i=1}^{\infty}$ *be given as described in Fig. 1. If the sequence* $\{T_i\}_{i=0}^{\infty}$ *is bounded, then the sequence* $\{H_i\}_{i=0}^{\infty}$ *converges to* V.

Proof. Suppose the sequence $\{T_i\}_{i=0}^{\infty}$ is bounded, i.e., $\forall i \in \mathbb{N} : |T_i| \leq \Theta$ for some constant Θ. Because $\forall i \in \mathbb{N} : B_i \geq 2^i$, then $|T_i|/B_i \leq \Theta/2^i$ and therefore $\lim_{i\to\infty} T_i/B_i = 0$. Now $H_i = V - T_i/B_i$, and so

$$\begin{aligned}
\lim_{i\to\infty} H_i &= \lim_{i\to\infty}(V - T_i/B_i)\\
&= \lim_{i\to\infty} V - \lim_{i\to\infty} T_i/B_i = V.
\end{aligned}$$

□

4 On-the-Fly Technique

On-the-fly techniques were first introduced by Ercegovac and Lang [9], [10] and further studied by Frougny [11]. When the

[2] See the description of radix-conversion in [8].

$\beta_i A_{i-1}$	A_{i-1}				0	0	0	0
v_i	s	s	s	s	v	v	v	v
Sum when $s = 0$	A_{i-1}				v	v	v	v
Sum when $s = 1$	$A_{i-1} - 1$				v	v	v	v

Fig. 2. 1-bit overlap; $\beta_i A_{i-1} + v_i$ when $\mu_i = 4$ and $s \equiv \mathrm{signbit}(v_i)$.

technique applies, it offers an efficient way to accumulate the (integer) digits generated by a DSM. The binary on-the-fly technique can be described as follows. We assume integers are represented using two's complement notation, and that $\forall i \in \mathbb{N}^{>0} : \beta_i \equiv 2^{\mu_i}$ where each $\mu_i \in \mathbb{N}^{>0}$.

First, no accumulation is needed to form $H_1 = v_1$, nor is there any restriction placed on the magnitude of the digit v_1. Next, for $i \geq 2$, consider how the digit v_i is accumulated into H_{i-1} to form H_i

$$H_i \equiv H_{i-1} + \frac{v_i}{B_i}.$$

Adding v_i/B_i to H_{i-1} creates a carry chain whose length can be nearly the bit-width of H_{i-1}. The goal of the on-the-fly technique is to eliminate this addition and its associated carry chain.

The simplest form of the on-the-fly technique assumes that $\forall i \geq 2 : |v_i| < \beta_i$, so both v_i and $v_i - 1$ have $(\mu_i + 1)$-bit two's complement representations. For each $i \geq 2$,

$$B_i H_i = B_i H_{i-1} + v_i = \beta_i B_{i-1} H_{i-1} + v_i,$$

and so

$$A_i = \beta_i A_{i-1} + v_i,$$

where $A_i \equiv B_i H_i$. Consider Fig. 2 which illustrates the alignment of $\beta_i A_{i-1}$ and the sign-extended form of v_i when $\mu_i = 4$; note the 1 bit overlap between the leading (sign) bit of v_i and the trailing bit of A_{i-1}. When interpreted as a two's complement integer, the value of the bits of the sign-extended form of v_i that overlap A_{i-1} is either -1 or 0. From this observation we draw the following conclusions:

- when $v_i \in \mathbb{N}$: $s = 0$ and A_i is formed by concatenating the bits of A_{i-1} and the μ_i trailing bits of v_i, and
- when $v_i < 0$: $s = 1$ and A_i is formed by concatenating the bits of $A_{i-1} - 1$ with the μ_i trailing bits of v_i.

Consequently, if A_{i-1} and $A'_{i-1} \equiv A_{i-1} - 1$ are given, then A_i can be formed by appending the μ_i trailing bits of v_i to a selection of either A_{i-1} or A'_{i-1}. An analogous argument applies to the formation of $A'_i \equiv A_i - 1$ because

$$A'_i \equiv A_i - 1 = \beta_i A_{i-1} + v_i - 1 = \beta_i A_{i-1} + w_i,$$

where we recall that $w_i \equiv v_i - 1$ also has a $(\mu_i + 1)$-bit two's complement representation. In summary,

$$A_i = \begin{cases} \mathrm{concatenate}(A_{i-1}, T_{\mu_i}(v_i)) & \text{if } v_i \in \mathbb{N} \\ \mathrm{concatenate}(A'_{i-1}, T_{\mu_i}(v_i)) & \text{if } v_i < 0 \end{cases} \text{ and}$$

$$A'_i = \begin{cases} \mathrm{concatenate}(A_{i-1}, T_{\mu_i}(w_i)) & \text{if } w_i \in \mathbb{N} \\ \mathrm{concatenate}(A'_{i-1}, T_{\mu_i}(w_i)) & \text{if } w_i < 0. \end{cases}$$

where $T_\mu(z)$ consist of the trailing μ bits of the two's complement representation of the integer z.

$\beta_i A_{i-1}$	A_{i-1}				0	0	0	0
v_i	s	s	s	v	v	v	v	v

Fig. 3. 2-bit overlap; $\beta_i A_{i-1} + v_i$ when $\mu_i = 4$ and $s \equiv \mathrm{signbit}(v_i)$.

```
procedure DSM_PROXY($V, \{\psi_i\}_{i=0}^{\infty}$)
  $(B_0, H_0, T_0) := (1, 0, V)$
  for $i := 0, 1, 2, \ldots$ do
    {Invariant: $V = H_i + T_i/B_i$}
    $T_i^p := (1 + \psi_i) T_i$
    $v_{i+1} := \mathrm{DSF}_{i+1}(\beta_{i+1} T_i^p)$
    $B_{i+1} := \beta_{i+1} B_i$
    $H_{i+1} := H_i + v_{i+1}/B_{i+1}$
    $T_{i+1} := \beta_{i+1} T_i - v_{i+1}$
  end for
end procedure
```

Fig. 4. DSM using a Proxy with $V \in \mathbb{R}^{\geq 0}$, $\forall i \in \mathbb{N}^{>0} : \beta_i \geq 2$, and $\forall i \in \mathbb{N}^{>0} : \mathrm{DSF}_i \in \mathrm{RNI}(\Omega_i)$.

This argument can be generalized in several ways. Consider, for example, the case where the digits cover the wider range $\forall i \geq 2 : |v_i| < 2\beta_i - 1$. In this case, because $(\mu_i + 2)$-bit two's complement integers range from $-2\beta_i$ to $2\beta_i - 1$ inclusively, each of the integers $\{v_i - 2, v_i - 1, v_i, v_i + 1\}$ has a $(\mu_i + 2)$-bit two's complement representation. Fig. 3 illustrates the addition of one of these four integers to $\beta_i A_{i-1}$; note the 2-bit overlap between that integer and $\beta_i A_{i-1}$. The integer described by the bits in the overlap of the sign-extended form of the integer and $\beta_i A_{i-1}$ ranges from -2 to 1, inclusively. Therefore, because

$$A_i + k = \beta_i A_{i-1} + (v_i + k) \quad \text{for} \quad k \in \{-2, -1, 0, 1\},$$

we can form any one of the values $\{A_i - 2, A_i - 1, A_i, A_i + 1\}$ by adding the corresponding integer $\{v_i - 2, v_i - 1, v_i, v_i + 1\}$ to $\beta_i A_{i-1}$. For example, to form $A_i - 2$ add $z_i \equiv v_i - 2$ to $\beta_i A_{i-1}$. To perform this addition use the 2 leading bits of the $(\mu_i + 2)$-bit two's complement representation of z_i to select to which of $\{A_{i-1} - 2, A_{i-1} - 1, A_{i-1}, A_{i-1} + 1\}$ the trailing μ_i-bits of z_i are appended.

5 DSM Using a Proxy

Procedure DSM_BASIC is not effective for several reasons.

First, the value of V is used to initialize T_i. That's acceptable for recoding, where the algorithm converts the value of V in one form (say, binary) into another form (say, decimal). It's also acceptable in an analysis of the algorithm. It is not acceptable when actually performing a division or square root because it presupposes that the result of the computation is known before the algorithm starts.

Second, when the algorithm is applied to division or square root, the computation of the tails T_i involves a nontrivial division. For example, with the invariant written as $\forall i \in \mathbb{N} : T_i = B_i(V - H_i)$, it is simple to derive for the division problem $V \equiv X/Y$ that

$$\forall i \in \mathbb{N} : T_i Y = B_i(X - H_i Y),$$

and for the square root problem $V \equiv \sqrt{X}$ that

$$\forall i \in \mathbb{N} : T_i(V + H_i)/2 = B_i(X - H_i^2)/2.$$

In each of these equalities, the right-hand side can be computed via addition and multiplication of known finite precision values and the finite precision estimate H_i of V. However, given these right-hand sides, an unavoidable nontrivial division is required to determine the values of T_i.

Procedure DSM_BASIC determines the next digit v_{i+1} by approximately rounding $\beta_{i+1} T_i$ to an integer. It is plausible, then, that v_{i+1} can be determined using an accurate[3] proxy T_i^p for T_i.

[3] The accuracy of an approximation is measured by its relative error. The relative error of an approximation A' of $A \neq 0$ is $|\psi|$ where $A' = (1 + \psi)A$.

Procedure DSM_PROXY in Fig. 4 is a template for a DSM that uses a proxy T_i^p for T_i; it reduces to DSM_BASIC when $\forall i \in \mathbb{N} : \psi_i = 0$.

We make two assumptions about the proxies $\{T_i^p\}_{i=0}^{\infty}$.

- For analysis: The proxy T_i^p can be expressed as $T_i^p = (1 + \psi_i)T_i$; if $T_i \neq 0$ then $|\psi_i|$ is the relative error in the approximation of T_i by the proxy T_i^p.
- For implementation: The proxy T_i^p can be computed without knowledge of the exact values of V and T_i. When this assumption is satisfied, occurrences of V and T_i in DSM_PROXY can be eliminated. Examples of this elimination are presented in the following sections.

For DSM_PROXY, the sequences $\{\mathrm{DSF}_i\}_{i=1}^{\infty}$ and $\{\beta_i\}_{i=1}^{\infty}$ are considered to be fixed and to honor the restrictions stated in the caption. We also suppose that ψ_i depends on V, T_i, and H_i; the dependence on H_i can be eliminated by applying the invariant $H_i = V - T_i/B_i$. In summary, T_{i+1} can be determined from just V and T_i.

To reduce the notational load, the dependence of T_i and T_i^p on V is represented implicitly.

Theorem 5.1 (Proxy Theorem). *In DSM_PROXY suppose that for some $V \in \mathbb{R}^{\geq 0}$ the sequence $\{\psi_i\}_{i=0}^{\infty}$ satisfies $\forall i \in \mathbb{N}, t \in \mathbb{R} : |\psi_i(V, t)| \leq \Psi_i(V, |t|)$ where Ψ_i is a non-decreasing function of its second argument. Then for that V,*

$$\forall i \in \mathbb{N} : (|T_i| \leq \tau_i(V)) \wedge \left(|T_i^p| \leq \tau_i^p(V)\right),$$

where $\tau_i, \tau_i^p : \mathbb{R}^{\geq 0} \to \mathbb{R}$ are defined as

$$\begin{aligned}
&\tau_0(u) \equiv u, \\
&\forall i \in \mathbb{N} : \tau_{i+1}(u) \equiv \beta_{i+1}\Psi_i(u, \tau_i(u))\tau_i(u) + \Omega_{i+1}, \text{ and} \\
&\forall i \in \mathbb{N} : \tau_i^p(u) \equiv (1 + \Psi_i(u, \tau_i(u)))\tau_i(u).
\end{aligned}$$

Proof. Suppose that $V \in \mathbb{R}^{\geq 0}$ and the sequence $\{\psi_i\}_{i=0}^{\infty}$ satisfies $\forall i \in \mathbb{N}, t \in \mathbb{R} : |\psi_i(V, t)| \leq \Psi_i(V, |t|)$ where Ψ_i is a non-decreasing function of its second argument.

We inductively prove that $\forall i \in \mathbb{N} : |T_i| \leq \tau_i(V)$ as follows. The base case is true $|T_0| = V = \tau_0(V)$. For the inductive step assume that $|T_i| \leq \tau_i(V)$ for some $i \in \mathbb{N}$. We know $T_i^p = (1 + \psi_i(V, T_i))T_i$, so application of the triangle inequality yields:

$$\begin{aligned}
|T_{i+1}| &= |\beta_{i+1}T_i - v_{i+1}| \\
&= |\beta_{i+1}T_i - \mathrm{DSF}_{i+1}(\beta_{i+1}T_i^p)| \\
&= |\beta_{i+1}T_i - (\beta_{i+1}T_i^p - \mathrm{coDSF}_{i+1}(\beta_{i+1}T_i^p))| \\
&= |\beta_{i+1}(T_i - T_i^p) + \mathrm{coDSF}_{i+1}(\beta_{i+1}T_i^p)| \\
&= |-\beta_{i+1}\psi_i(V, T_i)T_i + \mathrm{coDSF}_{i+1}(\beta_{i+1}T_i^p)| \\
&\leq \beta_{i+1}|\psi_i(V, T_i)||T_i| + \Omega_{i+1}.
\end{aligned}$$

Next, apply the assumption that $|\psi_i(V, t)| \leq \Psi_i(V, |t|)$, where Ψ_i is a non-decreasing function of its second argument, to continue this inequality as follows.

$$\begin{aligned}
|T_{i+1}| &\leq \beta_{i+1}|\psi_i(V, T_i)||T_i| + \Omega_{i+1} \\
&\leq \beta_{i+1}\Psi_i(V, |T_i|)|T_i| + \Omega_{i+1} \\
&\leq \beta_{i+1}\Psi_i(V, \tau_i(V))\tau_i(V) + \Omega_{i+1} \equiv \tau_{i+1}.
\end{aligned}$$

This completes the induction.

With the bounds on $\forall i \in \mathbb{N} : |T_i| \leq \tau_i(V)$ established, the bounds on $\forall i \in \mathbb{N} : |T_i^p|$ are obtained as follows. For each $i \in \mathbb{N}$:

```
procedure DSM_DIV($X, Y$)
    $(B_0, H_0, R_0) := (1, 0, X)$
    for $i := 0, 1, 2, \ldots$ do
        {Invariant: $X = H_iY + R_i/B_i$}
        $T_i^p := g(Y)R_i$
        $v_{i+1} := \mathrm{DSF}_{i+1}(\beta_{i+1}T_i^p)$
        $B_{i+1} := \beta_{i+1}B_i$
        $H_{i+1} := H_i + v_{i+1}/B_{i+1}$
        $R_{i+1} := \beta_{i+1}R_i - v_{i+1}Y$
    end for
end procedure
```

Fig. 5. Division DSM using a Proxy with $X \in [1/2, 1)$, $Y \in [1, 2)$, $V \equiv X/Y$, $\forall i \in \mathbb{N}^{>0} : \beta_i \geq 2$, and $\forall i \in \mathbb{N}^{>0} : \mathrm{DSF}_i \in \mathrm{RNI}(\Omega_i)$.

$$\begin{aligned}
|T_i^p| &= |(1 + \psi_i(V, T_i))T_i| \\
&\leq (1 + |\psi_i(V, T_i)|)|T_i| \\
&\leq (1 + \Psi_i(V, |T_i|))|T_i| \\
&\leq (1 + \Psi_i(V, \tau_i(V)))\tau_i(V) \equiv \tau_i^p(V).
\end{aligned}$$

□

Definition 5.2. *Let $\mathbb{P}$ be the subset of functions $\mathbb{R}^{>0} \to \mathbb{R}$ for which $p \in \mathbb{P}$ whenever $p(V)$ is a finite sum of terms of the form cV^n where $c \in \mathbb{R}^{\geq 0}$ and $n \in \mathbb{Z}$.*

Each $p \in \mathbb{P}$ is a convex function because on $\mathbb{R}^{>0}$ its second derivative is non-negative. Among the elements of $\mathbb{P}$ are each non-negative constant function as well as the identity function ν where $\forall u \in \mathbb{R}^{>0} : \nu(u) = u$. We also have these closure properties: for $p, q \in \mathbb{P}$ the functions p/ν, $p + q$, $pq \in \mathbb{P}$.

Corollary 5.3. *Let the assumptions of Theorem 5.1 hold for every $V \in \mathbb{R}^{>0}$. If*

$$\forall i \in \mathbb{N}, p \in \mathbb{P} : \Phi_i(p) \in \mathbb{P}.$$

where $\Phi_i : \mathbb{P} \to \mathbb{R}^{>0} \to \mathbb{R}$ is defined as

$$\forall p \in \mathbb{P}, u \in \mathbb{R}^{>0} : \Phi_i(p)(u) = \Psi_i(u, p(u)).$$

then for any closed subinterval $[a, b]$ of $\mathbb{R}^{>0}$,

$$\begin{aligned}
&\forall i \in \mathbb{N}, u \in [a, b]; \tau_i(u) \leq t_i \equiv \max(\tau_i(a), \tau_i(b)), \\
&\forall i \in \mathbb{N}, u \in [a, b]; \tau_i^p(u) \leq t_i^p \equiv \max(\tau_i^p(a), \tau_i^p(b)).
\end{aligned}$$

Proof. Let the assumptions of this corollary hold. We first prove inductively that $\tau_i \in \mathbb{P}$ for each $i \in \mathbb{N}$. The base case is true because $\tau_0 = \nu \in \mathbb{P}$. For the inductive step let $\tau_i \in \mathbb{P}$ for some $i \in \mathbb{N}$. By assumption $\Phi_i(\tau_i) \in \mathbb{P}$, so by the closure properties $\tau_{i+1} = \beta_{i+1}\Phi_i(\tau_i)\tau_i + \Omega_{i+1} \in \mathbb{P}$, and this completes the inductive argument. Next, consider τ_i^p for any $i \in \mathbb{N}$. By assumption $\Phi_i(\tau_i) \in \mathbb{P}$ because $\tau_i \in \mathbb{P}$, so by the closure properties $\tau_i^p = (1 + \Phi_i(\tau_i))\tau_i \in \mathbb{P}$.

Let $[a, b]$ be a closed subinterval of $\mathbb{R}^{>0}$. Because functions in $\mathbb{P}$ are convex, we know [12] that τ_i and τ_i^p attain their maximum on $[a, b]$ at either a or b. □

Combining the Theorem 3.2 with Corollary 5.3 yields for each $i \in \mathbb{N}$ and $V \in [a, b]$ that

$$|T_i| \leq t_i \quad \text{and} \quad |v_{i+1}| \leq \lfloor \beta_{i+1}t_i^p + \Omega_{i+1} \rfloor.$$

The formalization of the Proxy Theorem using the HOL Light theorem prover is presented in Section 10.

```
procedure DSM_SQRT($X$)
  $(B_0, H_0, R_0) := (1, 0, X/2)$
  for $i := 0, 1, 2, \ldots$ do
    {Invariant: $X = H_i^2 + 2R_i/B_i$}
    $T_i^p := \mu_i g(X) R_i$
    $v_{i+1} := \mathrm{DSF}_{i+1}(\beta_{i+1} T_i^p)$
    $B_{i+1} := \beta_{i+1} B_i$
    $H_{i+1} := H_i + v_{i+1}/B_{i+1}$
    $R_{i+1} := \beta_{i+1} R_i - v_{i+1}(H_{i+1} + H_i)/2$
  end for
end procedure
```

Fig. 6. Square root DSM using a proxy with $X \in [1/4, 1)$, $V \equiv \sqrt{X}$, $\forall i \in \mathbb{N}^{>0} : \beta_i \geq 2$, and $\forall i \in \mathbb{N}^{>0} : \mathrm{DSF}_i \in \mathrm{RNI}(\Omega_i)$.

6 DSM for Division

As discussed in Section 2, we consider the computation of $V \equiv X/Y$ where $X \in [1/2, 1)$ and $Y \in [1, 2)$. Procedure DSM_DIV in Fig. 5 is an effective DSM that computes V; it uses an approximation $g(Y)$ of $1/Y$ obtained from, say, a lookup table. (Microprocessors often have an approximate reciprocal instruction.) The relative error in this approximation at Y is $|\sigma(Y)|$ where $\sigma : [1, 2) \to \mathbb{R}$ is defined so that

$$\forall Y \in [1, 2) : g(Y) \equiv (1 + \sigma(Y))/Y.$$

We assume $\forall Y \in [1, 2) : |\sigma(Y)| \leq \Sigma$ for some constant Σ.

Reintroduce into DSM_DIV the recursive computation of T_i as in DSM_PROXY, and with it the invariant $\forall i \in \mathbb{N} : V = H_i + T_i/B_i$. As described in Section 5, from this invariant we find that

$$\forall i \in \mathbb{N} : T_i Y = \underbrace{B_i(X - H_i Y)}_{\tilde{R}_i}.$$

The $\tilde{R}_i$ are called *partial remainders* for division and admit, for all $i \in \mathbb{N}$, the identity:

$$\begin{aligned}\tilde{R}_{i+1} - \beta_{i+1}\tilde{R}_i &= B_{i+1}(X - H_{i+1}Y) - \beta_{i+1}B_i(X - H_i Y)\\ &= -B_{i+1}(H_{i+1} - H_i)Y\\ &= -v_{i+1}Y.\end{aligned}$$

We conclude that the partial remainders $\tilde{R}_i$ form one solution of the recurrence

$$\begin{aligned}&\tilde{R}_0 = X,\\ &\forall i \in \mathbb{N} : \tilde{R}_{i+1} = \beta_{i+1}\tilde{R}_i - v_{i+1}Y.\end{aligned}$$

The R_i computed by DSM_DIV form another solution of this recurrence. Because this recurrence has a unique solution, we conclude that $\forall i \in \mathbb{N} : \tilde{R}_i = R_i$.

The approximate identity $g(Y)Y \approx 1$ allows division by Y to be replaced, approximately, by multiplication by $g(Y)$. Recall that $\forall i \in \mathbb{N} : T_i Y = R_i$, so the proxy T_i^p for T_i is

$$\forall i \in \mathbb{N} : T_i^p \equiv g(Y)R_i.$$

A short computation shows that

$$\forall i \in \mathbb{N} : T_i^p = g(Y)R_i = g(Y)YT_i = (1 + \sigma(Y))T_i,$$

so the Proxy Theorem 5.1 can be applied with $\forall i \in \mathbb{N} : \psi_i(V, t) \equiv \sigma(Y)$ and $\forall i \in \mathbb{N} : \Psi_i(V, \tau) \equiv \Sigma$ because

$$\forall i \in \mathbb{N} : |\psi_i(V, t)| \equiv |\sigma(Y)| \leq \Sigma \equiv \Psi_i(V, |t|).$$

Clearly $\forall i \in \mathbb{N}, p \in \mathbb{P} : \Phi_i(p) = \Sigma \in \mathbb{P}$, so Corollary 5.3 applies. We conclude that $\forall i \in \mathbb{N} : |T_i| \leq t_i \equiv \tau_i(1)$ and $\forall i \in \mathbb{N} : |T_i^p| \leq t_i^p \equiv \tau_i^p(1)$ because each τ_i and τ_i^p is a non-negative increasing linear function on $[1/4, 1)$.

7 DSM for Square Root

As discussed in Section 2, we consider the computation of $V \equiv \sqrt{X}$ for $X \in [1/4, 1)$. Procedure DSM_SQRT in Fig. 6 is an effective DSM that computes V; it uses an approximation $g(X)$ of $1/\sqrt{X}$. (Microprocessors often have an approximate reciprocal square root instruction.) The relative error in this approximation at X is $|\sigma(X)|$ where $\sigma : [1/4, 1) \to \mathbb{R}$ is defined so that

$$\forall X \in [1/4, 1) : g(X) \equiv (1 + \sigma(X))/\sqrt{X}.$$

We assume $\forall X \in [1/4, 1) : |\sigma(X)| \leq \Sigma$ for some constant Σ.

Reintroduce into DSM_SQRT the recursive computation of T_i as in DSM_PROXY, and with it the invariant $\forall i \in \mathbb{N} : V = H_i + T_i/B_i$. As described in Section 5, from this invariant we find that

$$\forall i \in \mathbb{N} : T_i(V + H_i)/2 = \underbrace{B_i(X - H_i^2)/2}_{\tilde{R}_i}.$$

The $\tilde{R}_i$ are called *partial remainders* for square root and admit, for all $i \in \mathbb{N}$, the identity:

$$\begin{aligned}\tilde{R}_{i+1} - \beta_{i+1}\tilde{R}_i &= B_{i+1}\frac{X - H_{i+1}^2}{2} - \beta_{i+1}B_i\frac{X - H_i^2}{2}\\ &= -B_{i+1}(H_{i+1} - H_i)\frac{H_{i+1} + H_i}{2}\\ &= -v_{i+1}\frac{H_{i+1} + H_i}{2}.\end{aligned}$$

We conclude that the partial remainders $\tilde{R}_i$ form one solution of the recurrence

$$\begin{aligned}&\tilde{R}_0 = X/2,\\ &\forall i \in \mathbb{N} : \tilde{R}_{i+1} = \beta_{i+1}\tilde{R}_i - v_{i+1}(H_{i+1} + H_i)/2.\end{aligned}$$

The R_i computed by DSM_SQRT form another solution of this recurrence. Because this recurrence has a unique solution, we conclude that $\forall i \in \mathbb{N} : \tilde{R}_i = R_i$.

The proxy T_i^p for T_i is obtained by dividing R_i by an approximation of $(V + H_i)/2$. We argue that the approximate identity $\forall i \in \mathbb{N} : \mu_i g(X)(V + H_i)/2 \approx 1$ holds where

$$\mu_i \equiv (\text{if } i = 0 \text{ then } 2 \text{ else } 1),$$

because $g(X)V \approx 1$, $H_0 = 0$, and we expect $\forall i \in \mathbb{N}^{>0} : H_i \approx V$. This approximate identity allows division by $(V + H_i)/2$ to be replaced with multiplication by $\mu_i g(X)$, so the proxy T_i^p for T_i is

$$\forall i \in \mathbb{N} : T_i^p \equiv \mu_i g(X) R_i.$$

(The invariant tells us that $T_i = 2B_i V$ when $(V + H_i)/2 = 0$.)

Let $X \in [1/4, 1)$ be fixed, so $V \equiv \sqrt{X} \in [1/2, 1)$. For any $i \in \mathbb{N}$ we know $(V + H_i)/2 = V - T_i/(2B_i) = V(1 - T_i/(2VB_i))$ and $g(X)V = 1 + \sigma(X)$, so

$$\begin{aligned}T_i^p &\equiv \mu_i g(X) R_i\\ &= \mu_i g(X)((V + H_i)/2)T_i\\ &= \mu_i g(X)V(1 - T_i/(2VB_i))T_i\\ &= \mu_i(1 + \sigma(X))(1 - T_i/(2VB_i))T_i.\end{aligned}$$

Therefore, $T_i^p = (1 + \psi_i(V, T_i))T_i$ where

$$\psi_i(V, t) \equiv \sigma(X) - \begin{cases} 0 & \text{if } i = 0\\ (1 + \sigma(X))(t/(2VB_i)) & \text{if } i > 0, \end{cases}$$

because $\mu_0(1 - T_0/(2VB_0)) = 1$, and so the Proxy Theorem 5.1 can be applied using

TABLE 1
DSM (Using a Proxy) for Division and Square Root with $\Sigma = 2^{-9}$ and $\Omega = 5/8$

Division							V = 1/4			V = 1		
i	$log_2(\beta_i)$	β_i	B_i	t_i	t_i^p	Digit Bound	$\tau_i(V)$	$\Phi_i(\tau_i)(V)$	$\tau_i^p(V)$	$\tau_i(V)$	$\Phi_i(\tau_i)(V)$	$\tau_i^p(V)$
0			1.00E+00	1.0000	1.0020		0.2500	0.0020	0.2505	1.0000	0.0020	1.0020
1	7	128	1.28E+02	0.8750	0.8767	128	0.6875	0.0020	0.6888	0.8750	0.0020	0.8767
2	7	128	1.64E+04	0.8438	0.8454	112	0.7969	0.0020	0.7984	0.8438	0.0020	0.8454
3	7	128	2.10E+06	0.8359	0.8376	108	0.8242	0.0020	0.8258	0.8359	0.0020	0.8376
4	7	128	2.68E+08	0.8340	0.8356	107	0.8311	0.0020	0.8327	0.8340	0.0020	0.8356

Division							V = 1/4			V = 1		
i	$log_2(\beta_i)$	β_i	B_i	t_i	t_i^p	Digit Bound	$\tau_i(V)$	$\Phi_i(\tau_i)(V)$	$\tau_i^p(V)$	$\tau_i(V)$	$\Phi_i(\tau_i)(V)$	$\tau_i^p(V)$
0			1.00E+00	1.0000	1.0020		0.2500	0.0020	0.2505	1.0000	0.0020	1.0020
1	7	128	1.28E+02	0.8750	0.8767	128	0.6875	0.0020	0.6888	0.8750	0.0020	0.8767
2	5	32	4.10E+03	0.6797	0.6810	28	0.6680	0.0020	0.6693	0.6797	0.0020	0.6810
3	7	128	5.24E+05	0.7949	0.7965	87	0.7920	0.0020	0.7935	0.7949	0.0020	0.7965
4	7	128	6.71E+07	0.8237	0.8253	102	0.8230	0.0020	0.8246	0.8237	0.0020	0.8253

Square Root							V = 1/2			V = 1		
i	$log_2(\beta_i)$	β_i	B_i	t_i	t_i^p	Digit Bound	$\tau_i(V)$	$\Phi_i(\tau_i)(V)$	$\tau_i^p(V)$	$\tau_i(V)$	$\Phi_i(\tau_i)(V)$	$\tau_i^p(V)$
0			1.00E+00	1.0000	1.0020		0.5000	0.0020	0.5010	1.0000	0.0020	1.0020
1	7	128	1.28E+02	0.8750	0.8797	128	0.7500	0.0078	0.7559	0.8750	0.0054	0.8797
2	7	128	1.64E+04	1.3761	1.3789	113	1.3761	0.0020	1.3789	1.2273	0.0020	1.2298
3	7	128	2.10E+06	0.9838	0.9858	177	0.9838	0.0020	0.9858	0.9377	0.0020	0.9396
4	7	128	2.68E+08	0.8710	0.8727	126	0.8710	0.0020	0.8727	0.8595	0.0020	0.8611

Square Root							V = 1/2			V = 1		
i	$log_2(\beta_i)$	β_i	B_i	t_i	t_i^p	Digit Bound	$\tau_i(V)$	$\Phi_i(\tau_i)(V)$	$\tau_i^p(V)$	$\tau_i(V)$	$\Phi_i(\tau_i)(V)$	$\tau_i^p(V)$
0			1.00E+00	1.0000	1.0020		0.5000	0.0020	0.5010	1.0000	0.0020	1.0020
1	7	128	1.28E+02	0.8750	0.8797	128	0.7500	0.0078	0.7559	0.8750	0.0054	0.8797
2	5	32	4.10E+03	0.8128	0.8145	28	0.8128	0.0022	0.8145	0.7756	0.0020	0.7772
3	7	128	5.24E+05	0.8489	0.8505	104	0.8489	0.0020	0.8505	0.8283	0.0020	0.8299
4	7	128	6.71E+07	0.8374	0.8390	109	0.8374	0.0020	0.8390	0.8322	0.0020	0.8338

$$\Psi_i(V, |t|) \equiv \Sigma + \begin{cases} 0 & \text{if } i = 0 \\ (1+\Sigma)(|t|/(2VB_i)) & \text{if } i > 0. \end{cases}$$

Note that the first term Σ also occurs in Ψ_i for division. Clearly $\forall i \in \mathbb{N}, p \in \mathbb{P} : \Phi_i(p) \in \mathbb{P}$, so Corollary 5.3 applies and we conclude that $|T_i| \leq t_i \equiv \max(\tau_i(1/2), \tau_i(1))$ and $|T_i^p| \leq t_i^p \equiv \max(\tau_i^p(1/2), \tau_i^p(1))$.

8 Application

The results displayed in Table 1 describe the evolution of the bounds on the tails, tail proxies, and digits for the DSM algorithms for division and square root presented in the previous two sections. In this table the reciprocal and reciprocal root approximations are characterized by $\Sigma \equiv 2^{-9}$, and all digit selection functions belong to $\mathrm{RNI}(\Omega)$ for $\Omega \equiv 5/8$. (The *PN*^{2} or *PNQ* recoders discussed in [13] provide such digit selection functions.)

The table displays results for two choices of β-sequence:

- $\{\beta_1, \beta_2, \beta_3, \beta_4\} \equiv \{2^7, 2^7, 2^7, 2^7\}$, and
- $\{\beta_1, \beta_2, \beta_3, \beta_4\} \equiv \{2^7, 2^5, 2^7, 2^7\}$.

for each of division and square root. For each of these we obtain from Corollary 5.3, with ν the identity function, that

$$\tau_0 \equiv \nu,$$
$$\forall i \in \mathbb{N} : \tau_{i+1} \equiv \beta_{i+1}\Phi_i(\tau_i)\tau_i + \Omega,$$
$$\forall i \in \mathbb{N} : \tau_i^p \equiv (1 + \Phi_i(\tau_i))\tau_i,$$

where for division

$$\forall \tau \in \mathbb{P} : \Phi_i(\tau) \equiv \Sigma,$$

while for square root

$$\forall \tau \in \mathbb{P} : \Phi_i(\tau) \equiv \begin{cases} \Sigma & \text{if } i = 0 \\ \Sigma + (1+\Sigma)\tau/(2\nu B_i) & \text{if } i > 0. \end{cases}$$

For any given value of V, we know the value of τ_0 and so we can compute $\Phi_0(\tau_0)(V)$ and then $\tau_0^p(V)$. This pattern is repeated for $i = 1, 2, 3, 4$ in succession; compute $\tau_i(V)$, then $\Phi_i(\tau_i)(V)$ and $\tau_i^p(V)$. From Corollary 5.3 we obtain

$$\forall i \in \mathbb{N}, V \in [a,b]; \tau_i(V) \leq t_i \equiv \max(\tau_i(a), \tau_i(b)), \text{ and}$$
$$\forall i \in \mathbb{N}, V \in [a,b]; \tau_i^p(V) \leq t_i^p \equiv \max(\tau_i^p(a), \tau_i^p(b)),$$

where $[a,b] \equiv [1/4, 1]$ for division and $[a,b] \equiv [1/2, 1]$ for square root. Finally, for $i = 1, 2, 3, 4$:

$$|T_i| \leq t_i, \quad \text{and} \quad |v_i| \leq \lfloor \beta_i t_{i-1}^p + \Omega \rfloor.$$

Observe that, for square root, the first β-sequence leads to an upper bound on $|T_2|$ that is larger than 1, and so the bound on $|v_3|$ is larger than $2^{\beta_3} = 2^7 = 128$. For the second β-sequence, obtained from the first β-sequence by decreasing β_2 from 2^7 to 2^5, we find that $|v_i| < 2^{\beta_i}$ for $2 \leq i \leq 4$ as well as $|T_4| < 1$; so the simplest form of on-the-fly accumulation of the digits can be applied. The reason why the reduction of β_2 from 2^7 to 2^5 is effective can be explained by the fact that

$$\Phi_1(\tau_1) = \Sigma + (1+\Sigma)\tau_1/(2\nu\beta_1),$$

and so

$$\tau_2 = \beta_2\Sigma + \Omega_2 + \beta_2(1+\Sigma)\tau_1/(2\nu\beta_1).$$

From the corresponding example for division we know $\beta_2\Sigma + \Omega_2 = 1/4 + 5/8 = 7/8$ when $\beta_2 = 2^7$. The third term contains the ratio β_2/β_1, so when β_2 is reduced from 2^7 to 2^5 the contribution of this third term is reduced by a factor of 4.

We performed additional experiments using a spreadsheet implementation of the DSM for division and square root that expand on the results presented in Table 1. For specified values of the inputs (X and Y for division, X for square root), the spreadsheet computed the slack $s_i \equiv v_i^{max} - |v_i|$ where v_i^{max} is the upper bound on $|v_i|$ as discussed at the end of Section 5. The spreadsheet's optimizer was used to determine inputs that made s_i small, i.e., made $|v_i|$ close to v_i^{max}. For both division and square root, and for each $i \in \{1, 2, 3, 4\}$, the optimizer was able to find inputs that made $|v_i|$ at least 96 percent of v_i^{max}.

9 Conclusion

This paper has not assumed special properties of the digit selection functions or reciprocal approximations. Nor has it described specific digit selection functions; however see [4], [5], [13], [14], [15].

The analysis presented here extends to higher roots. For example, for the cube root $V = X^{1/3}$, from $T_i = B_i(V - H_i)$ it follows that

$$T_i(V^2 + VH_i + H_i^2)/3 = B_i(X - H_i^3)/3.$$

The partial remainders $R_i \equiv B_i(X - H_i^3)/3$ satisfy a two-term recurrence. Also, if $\nu_i = (\text{if i} == 0 \text{ then } 3 \text{ else } 1)$ and $g(X) \approx X^{-2/3}$, then $T_i^p \equiv \nu_i g(X) R_i$ is a natural choice as the proxy for T_i because $\nu_i g(X)(V^2 + VH_i + H_i^2)/3 \approx 1$.

Prescaled division, first presented by Svoboda [16], is also covered by the analysis presented here. Prescaled division computes $X' \equiv g(Y)X$ and $Y' \equiv g(Y)Y = 1 + \sigma(Y)$ before the for-loop; note that $X'/Y' = X/Y$. Inside the for-loop, the expressions

$$R_{i+1} = \beta_{i+1} R_i - v_{i+1} Y \quad \text{and} \quad X = H_i Y + R_i/B_i,$$

for the partial remainder and the invariant become, after multiplication by $g(Y)$,

$$\begin{aligned} R'_{i+1} &= \beta_{i+1} R'_i - v_{i+1} Y' \\ &= (\beta_{i+1} R'_i - v_{i+1}) - v_{i+1}\sigma(Y), \text{ and} \\ X' &= H_i Y' + R'_i/B_i, \end{aligned}$$

where $R'_i \equiv g(Y)R_i$. Note that $R'_0 \equiv X'$ and $T_i^p = R'_i$. The advantage of prescaled division is that, at a cost of two multiplications outside the for-loop, no multiplication inside the for-loop is needed to form the proxy T_i^p.

10 HOL Light Verification

The proofs of the Proxy Theorem, its Corollary and the applications to division and square root, including verification of some concrete error bounds for particular instances, have been formally checked using the HOL Light theorem prover [1]; for the details see [17]. The statement of the Proxy Theorem in HOL Light reads as follows:

```
let THEOREM_V_1 = prove
('!(V:real) (beta:num->real) (omega:num->real)
(DSF:num->real->real)   (B:num->real)   (H:num-
>real)
(v:num->real) (Tl:num->real) (Tp:num->real)
(PSI:num->real#real->real) (psi:num->real#
real->real)
(tau:num->real->real) (taup:num->real->real).

// Environmental assumptions including
nondecreasing
// property
&0 <= V /\
(!i. i >= 0 ==> beta i > &0) /\
(!i. i >= 1 ==> (!x. abs (x - DSF i x) <= omega i)) /\
(!i. i >= 0 ==>
   abs (psi i (V,Tl i)) <= PSI i (V,abs(Tl i))) /\
(!i x y. &0 <= x /\ x <= y ==>
   PSI i (V,x) <= PSI i (V,y)) /\

(!u. tau 0 u = u) /\
(!i u. tau (i + 1) u =
   beta (i + 1) * PSI i (u,tau i u) * tau i u +
   omega (i + 1)) /\
(!i u. taup i u = (&1 + PSI i (u,tau i u)) * tau i u) /\

// Computing recursively
B 0 = &1 /\ H 0 = &0 /\ Tl 0 = V /\
(!i. Tp i = (&1 + psi i (V,Tl i)) * Tl i) /\
(!i. v (i + 1) = DSF (i + 1) (beta (i + 1) * Tp i)) /\
(!i. B (i + 1) = beta (i + 1) * B i) /\
(!i. H (i + 1) = H i + v (i + 1) / B (i + 1)) /\
(!i. Tl (i + 1) = beta (i + 1) * Tl i - v (i + 1))

// Conclude loop invariant and bounds.
==> (!i. V = H i + Tl i / B i) /\
    (!i. i >= 0 ==> abs(Tl i) <= tau i V) /\
    (!i. i >= 0 ==> abs(Tp i) <= taup i V)',
```

In this statement the ampersand '&' is the injection from natural numbers to reals, so can more or less be ignored. The variable names in the theorem correspond closely to those in the informal statement except that T is here called `Tl` to avoid a clash with the constant `T` (Boolean 'true'). Subscripts are treated just as (additional) function arguments, as usual in formal treatments. Also, to avoid confusion, u instead of V is used as the argument of the functions $\tau_i(V)$ and $\tau_i^p(V)$.

The overall statement is an implication with a conjunct of hypotheses and a conjunct of three conclusions. The first few hypotheses correspond directly to those of the theorem (that `V` is nonnegative, that `DSF` returns a suitable approximation etc.) while the remainder give explicit hypothetical recursion equations for the defined/computed quantities. (Note that the existence of functions satisfying these recursion equations is easy to prove as they are all primitive recursive, so there is no danger of the theorem's holding vacuously — one could indeed state their existence as part of the conclusion.) The conclusion (consequent of the implication) is that the loop invariant and the bounds then hold as claimed. It is perhaps noteworthy/amusing that the main quantities (i.e., all variables except subscript i) are all taken to be arbitrary reals, and the fact that many quantities are integers plays no role in the formal proof.

Acknowledgments

We thank Ping Tak Peter Tang, John O'Leary, Simon Rubanovich, and David Russinoff for many conversations related to the design and validation of floating-point arithmetic units. We also thank the anonymous reviewers for allowing us to incorporate their insightful comments into this paper.

References

[1] J. Harrison, "HOL Light: A tutorial introduction," in *Proc. 1st Int. Conf. Formal Methods. Comput.-Aided Des.*, 1996, pp. 265–269.

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

[3] W. S. Briggs and D. W. Matula, "A 17× 69 bit multiply and add unit with redundant binary feedback and single cycle latency," in *Proc. 11th Symp. Comput. Arithmetic*, 1993, pp. 163–170.

[4] M. D. Ercegovac, T. Lang, and P. Montuschi, "Very-high radix division with prescaling and selection by rounding," *IEEE Trans. Comput.*, vol. 43, no. 8, pp. 909–918, Aug. 1994.

[5] T. Lang and P. Montuschi, "Very-high radix combined division and square root with prescaling and selection by rounding," in *Proc. 12th IEEE Symp. Comput. Arithmetic*, 1995, pp. 124–131.

[6] M. D. Ercegovac and J.-M. Muller, "Variable radix real and complex digit-recurrence division," in *Proc. 16th IEEE Int. Conf. Appl.-Specific Syst. Archit. Processors*, 2005, pp. 316–321.

[7] J. Harrison, “Formal verification of IA-64 division algorithms,” *Theorem Proving Higher Order Logics*, vol. 1869, pp. 233–251, 2000.

[8] D. E. Knuth, *Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Reading, MA, USA: Addison-Wesley Professional, Nov. 1997.

[9] M. D. Ercegovac and T. Lang, “On-the-fly conversion of redundant into conventional representations,” *IEEE Trans. Comput.*, vol. 36, no. 7, pp. 895–897, Jul. 1987.

[10] M. D. Ercegovac and T. Lang, “On-the-fly rounding for division and square root,” in *Proc. 9th IEEE Symp. Comput. Arithmetic*, 1989, pp. 169–173.

[11] C. Frougny, “On-the-fly algorithms and sequential machines,” *IEEE Trans. Comput.*, vol. 49, no. 8, pp. 859–863, Aug. 2000.

[12] C. Niculescu and L.-E. Persson, *Convex Functions and Their Applications: A Contemporary Approach*. Berlin, Germany: Springer Science & Business Media, 2006.

[13] M. Daumas and D. W. Matula,“Further reducing the redundancy of a notation over a minimally redundant digit set,” *J. VLSI Signal Process.*, vol. 33, no. 1, pp. 7–18, 2003.

[14] M. Daumas and D. Matula, “Recoders for partial compression and rounding,” Ecole Normale Supérieure de Lyon, Lyon Cedex 07, France, Research Rep. LIP-RR1997-01, 1997.

[15] M. D. Ercegovac and T. Lang, “On recoding in arithmetic algorithms,” in *Proc. Conf. Record 28th Asilomar Conf. Signals Syst. Comput.*, 1994, pp. 531–535.

[16] A. Svoboda, “An algorithm for division,” *Inf. Process. Mach.*, vol. 9, pp. 25–34, 1963.

[17] W. E. FergusonJr., J. Bingham, L. Erkök, J. R. Harrison, and J. Leslie-Hurd, “Digit serial methods with applications to division and square root - with mechanically checked correctness proofs,” pp. 1–32, Aug. 2017. [Online]. Available: https://arxiv.org/abs/1708.00140