

Single Precision Reciprocal by Multipartite Table Look-up

Peter Kornerup
University of Southern Denmark
Odense, Denmark

E-mail: kornerup@imada.sdu.dk

David W. Matula
Southern Methodist University
Dallas, Texas, USA

E-mail: matula@engr.smu.edu

Abstract—We develop the foundations for confirming monotonicity of a multi-term reciprocal function approximation. We introduce the concept of operand recoding to improve the accuracy of multipartite approximation. The results are applied to provide a proposed four-partite reciprocal implementation with total table size 27 Kbytes, that yields an IEEE standard, single precision format (24 bit) reciprocal instruction, that is a one-ulp monotonic reciprocal.

I. INTRODUCTION

There has been considerable investigation of bipartite and multipartite function approximations in the recent literature [1], [2], [3], [4], [5], [6], [7], [8], [9]. Bipartite reciprocal approximations have been employed for approximate (low precision) reciprocal instructions in commodity microprocessors, targeted at multimedia applications. The question of monotonicity of reciprocal approximations has been discussed in [5], [6]. In this paper we investigate the applicability of the multipartite approach to obtaining an IEEE single precision (24 bit) one-ulp monotonic reciprocal function.

Summary:

Given a single precision divisor, $y = 1.b_1b_2 \cdots b_{23} \in [1; 2 - 2^{-23}]$, we shall show that it is possible, by a multipartite table look-up method, to determine an approximate reciprocal value

$$\text{recip}_a(y) = 0.1a_2 \cdots a_{24} \underbrace{a_{25} \cdots a_{24+g}}_{\text{guardbits}} \in (\frac{1}{2}; 1) \quad (1)$$

or $\text{recip}_a(1) = 1.00 \cdots 0$, having relative error less than 2^{-25} . This is equivalent to having the absolute error bound:

$$\left| \frac{1}{y} - \text{recip}_a(y) \right| < \frac{1}{2y} 2^{-24}. \quad (2)$$

In Section II we develop the foundations of monotonic one-ulp reciprocal functions. In particular we introduce and prove a monotonicity theorem. Specifically, if $\text{recip}_a(y)$ satisfies (2), then the reciprocal function $\text{rn}_{24}(\text{recip}_a(y))$, obtained by rounding¹ such an approximate reciprocal function to nearest at the 24^{th} position, is a one-ulp monotonic single precision reciprocal.

For single precision division with dividend $x = 1.x_1x_2 \cdots x_{23}$, normalized so that $x \in [1; 2 - 2^{-23}]$, let q be the normalized (rational) exact quotient $q \in [1; 2 - 2^{-23}]$ given by

$$q = \begin{cases} \frac{2x}{y} & \text{for } x < y \\ \frac{x}{y} & \text{for } x \geq y. \end{cases}$$

For $\text{recip}_a(y)$ satisfying (1) and (2), let $q_a(x, y)$ be the normalized binary quotient approximation $q_a(x, y) = 1.q_1q_2 \cdots q_{47+g} \in [1; 2)$ determined by

$$q_a(x, y) = \begin{cases} 2x \cdot \text{recip}_a(y) & \text{for } x < y, \\ x \cdot \text{recip}_a(y) & \text{for } x > y, \\ 1 & \text{for } x = y. \end{cases}$$

Hence $|q - q_a(x, y)| < \frac{q}{2} 2^{-24} < \frac{1}{2} 2^{-23}$ from which it follows that $|q - \text{rn}_{23}(q_a(x, y))| < 2^{-23}$ and $|q - (\text{rd}_{23}(q_a(x, y)) + \frac{1}{2} 2^{-23})| < 2^{-23}$.

¹For $z \in [0; 2 - 2^{-j}]$, the *fixed-point round-down* $\text{rd}_j(z)$, *round-up* $\text{rn}_j(z)$, and *round-to-nearest (midpoint down)* $\text{rn}_j(z)$ roundings, each determine either the $(j+1)$ -bit unnormalized binary value $\frac{k}{2^j} = a_0.a_1a_2 \cdots a_j$ with $0 \leq k \leq 2^{j+1} - 1$. Specifically we have

$$\text{rn}_j(z) = \max_k \left\{ \frac{k}{2^j} \mid \frac{k}{2^j} \leq z + \frac{1}{2} \right\},$$

with similar expressions for $\text{ru}_j(z)$ and $\text{rd}_j(z)$. For normalized input $z \in [\frac{1}{2}, 1)$, the output is normalized, e.g., $\text{rn}_j(z) = 0.1a_2a_3 \cdots a_j$ or $\text{rn}_j(z) = 1$.

Note that $q^* = \text{rn}_{23}(q_a(x, y)) = 1.q_1^* \cdots q_{23}^*$ is a directed breakpoint in the sense that $\text{sign}(q^*y - x)$ allows q^* or $q^* \pm \text{ulp}$ to be correctly chosen as the precise round-down (or round-up) single precision result for division of x by y . Similarly, $q_m^* = \text{rd}_{23}(q_a(x, y)) + \frac{1}{2}2^{-23}$ is a round-to-nearest breakpoint, allowing $\text{sign}(q^*y - x)$ to dictate the correct round-to-nearest single precision division result.

Thus the multipartite table lookup procedure described here provides for implementing a one-ulp monotonic, single precision reciprocal function, without the need for a multiplier, and for obtaining a single precision division result, employing only two (dependent) single precision multiplications. Our suggested solution is a four-partite table lookup with total table size $26\frac{1}{2}$ Kbyte. These methods allow relatively low-power implementations of the SSE paired, single precision reciprocal and division instructions incorporated in current X-86 processors, targeted at low-power multimedia computations.

In Section III we review the fundamentals of bipartite table construction, and Section IV introduces the notion of operand partial recoding for constructing multipartite tables. In Section V we present a four-partite look-up table procedure for obtaining a single precision, one-ulp monotonic reciprocal function.

II. ULP-ACCURATE MONOTONIC RECIPROCAL FUNCTIONS

The reciprocal approximation $\text{recip}(y) = 0.1a_2a_3 \cdots a_j$ is termed a j -bit *one-ulp reciprocal* when $|\frac{1}{y} - \text{recip}(y)| < 2^{-j}$ for all normalized binary divisors $y \in [1; 2)$, and similarly is a j -bit $\frac{3}{4}$ -ulp reciprocal when $|\frac{1}{y} - \text{recip}(y)| < \frac{3}{4}2^{-j}$.

Observation 1: A j -bit one ulp reciprocal $\text{recip}(y)$ is either the round-up or round-down value of $\frac{1}{y}$ for all normalized binary divisors $y \in [1; 2)$. That is,

$$\text{recip}(y) \in \left\{ \text{rd}_j\left(\frac{1}{y}\right), \text{ru}_j\left(\frac{1}{y}\right) \right\} \text{ for all } y \in [1; 2),$$

with $\text{recip}(y)$ always being the j -bit value nearest in the direction of the approximation.

Note that a one ulp reciprocal is efficiently computable by first obtaining a multiple term reciprocal approximations $\text{recip}_a(y) = 0.1a_2a_3 \cdots a_ja_{j+1} \cdots a_{j+g}$, with

g guard bits $a_{j+1}a_{j+2} \cdots a_{j+g}$, that satisfies $|\frac{1}{y} - \text{recip}_a(y)| < \frac{1}{2}2^{-j}$. Then the guard bits are rounded off to obtain the j -bit one ulp reciprocal $\text{recip}(y) = \text{rn}_j(\text{recip}_a(y))$. Such one ulp reciprocals have applications as a short reciprocal in high radix division algorithms and as the approximate reciprocal function value for a reciprocal instruction implementation. For implementation of a one ulp reciprocal function as a reciprocal instruction, it is also desirable to investigate the monotonicity properties of such an approximate function.

Rounding Off Guard Bits - Monotonic Reciprocal Instruction: In the remainder of this section we focus on the important reciprocal function application where the (exact) inputs $y_i = 1.b_1b_2 \cdots b_{i-1}$, and (approximate) outputs $\text{recip}(y_i) = 0.1a_2a_3 \cdots a_i$ (or $\text{recip}(y_i) = 1$) are both i -bit normalized values with i too large for direct lookup to be practical, e.g. $i = 24$. In this case a multi-term computed reciprocal approximation with guard bits rounded off, to provide a one-ulp reciprocal is only guaranteed monotonic for y_i over the subinterval $[1; \sqrt{2})$. In particular, it can be shown that the output step size, for a one-ulp reciprocal for y_i over $[1; \sqrt{2})$, can vary from 0 to 3 ulps, and over $[\sqrt{2}; 2)$ the step size can be down by as much as two ulps, or reverse direction and be up by one ulp, contradicting monotonicity.

Figure 1(a) illustrates a 5-bit one ulp reciprocal $\text{ra}_5(\frac{1}{y_5})$ which systematically chooses the value of the pair $\{\text{rd}_5(\frac{1}{y_5}), \text{ru}_5(\frac{1}{y_5})\}$ that is at least one half-ulp away from $\frac{1}{y_5}$, i.e., where $\frac{1}{2}2^{-i} < |\frac{1}{y_5} - \text{ra}_5(\frac{1}{y_5})| < 2^{-5}$ for $y_i \in (1; 2)$. The step function graph in Figure 1(a) clearly illustrates that such a perverse, one-ulp reciprocal can have exaggerated variability in step size over $[1; \sqrt{2})$ and be non monotonic to the extent of virtual oscillation over $[\sqrt{2}; 2)$.

Note that computing $\text{recip}_a(y) = 0.1a_2a_3 \cdots a_i a_{i+1} \cdots$ satisfying $|\frac{1}{y} - \text{recip}_a(y)| < \frac{1}{4}2^{-i}$ results in $\text{recip}(y_i) = \text{rn}_i(\text{recip}_a(y))$ being a $\frac{3}{4}$ -ulp reciprocal. Figure 1(b) illustrates for $i = 5$ a 5-bit $\frac{3}{4}$ -ulp reciprocal function $\text{recip}(y_5)$ that chooses the farthest away of $\{\text{rd}_5(\frac{1}{y_5}), \text{ru}_5(\frac{1}{y_5})\}$ whenever the farthest yields $|\frac{1}{y_5} - \text{recip}(y_5)| < \frac{3}{4}2^{-5}$, and otherwise chooses the unique one satisfying the $\frac{3}{4}$ -ulp bound.

Lemma 2: For a normalized i -bit divisor $y_i =$

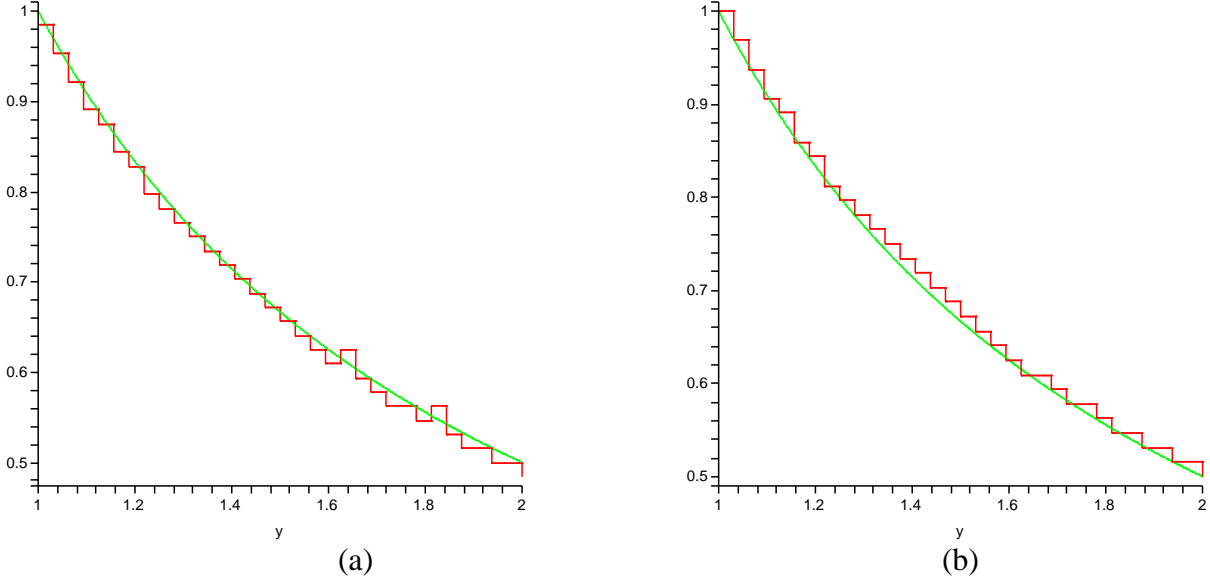


Fig. 1. (a): A 5-bit non-monotonic “round-away”, 1-ulp approximation, and (b): A $\frac{3}{4}$ -ulp monotonic approximation.

$1.b_1b_2\cdots b_{i-1}$, an i -bit $\frac{3}{4}$ -ulp reciprocal function is monotonic over the interval $[1; 2)$, and strictly monotonic over the portion $[1; \frac{2}{3}\sqrt{3} - 2^{-(i+1)})$.

Proof: For consecutive exact inputs $y_i, y_i + 2^{-(i-1)} \in [1; 2]$ the consecutive reciprocals decrease by $\frac{1}{y_i} - \frac{1}{y_i + 2^{-(i-1)}} = \frac{1}{y_i(y_i + 2^{-(i-1)})} 2^{-(i-1)} > \frac{1}{2} 2^{-i}$. Thus exact outputs decrease by at least one-half ulp of output. Suppose $\text{rd}_i(\frac{1}{y_i}) < \text{ru}_i(\frac{1}{y_i + 2^{-(i-1)}})$. Then the sum of the rounding errors satisfies $(\frac{1}{y_i} - \text{rd}_i(\frac{1}{y_i})) + (\text{ru}_i(\frac{1}{y_i + 2^{-(i-1)}}) - \frac{1}{y_i + 2^{-(i-1)}}) = (\frac{1}{y_i} - \frac{1}{y_i + 2^{-(i-1)}}) + (\text{ru}_i(\frac{1}{y_i + 2^{-(i-1)}}) - \text{rd}_i(\frac{1}{y_i})) > \frac{3}{2} 2^{-i}$. Hence at least one of the rounding errors would be greater than $\frac{3}{4}$ ulp, a contradiction. Thus $\text{rd}_i(\frac{1}{y_i}) \geq \text{ru}_i(\frac{1}{y_i + 2^{-(i-1)}})$ holds for all $y_i \in [1; 2)$, and a $\frac{3}{4}$ -ulp reciprocal function is monotonic (i.e., monotonically non increasing) over $[1; 2)$. Suppose $y_i \in [1; \frac{2}{3}\sqrt{3} - 2^{-i})$. Then $\frac{1}{y_i} - \frac{1}{y_i + 2^{-(i-1)}} > \frac{1}{(y_i + 2^{-i})^2} 2^{-(i-1)} > \frac{3}{4} 2^{-(i-1)} = \frac{3}{2} 2^{-i}$. If $\text{rd}_i(\frac{1}{y_i}) = \text{ru}_i(\frac{1}{y_i + 2^{-(i-1)}})$, the sum of rounding errors would be greater than $\frac{3}{2} 2^{-i}$, a contradiction. Thus a $\frac{3}{4}$ -ulp reciprocal is strictly monotonically decreasing for $y_i \in [1; \frac{2}{3}\sqrt{3} - 2^{-(i+1)})$. \square

In practice the “guarded” computation of a multi-term reciprocal approximation $\text{recip}_a(y_i)$ can often be shown to satisfy a maximum relative error bound for $y_i \in [1; 2)$, that is of the same order as the maximum absolute error bound for $y_i \in [1; 2)$. Importantly, obtaining just one extra bit of precision

in the relative error bound on $\text{recip}_a(y)$ before applying the final rounding is now shown sufficient to yield monotonicity.

Theorem 3 (Monotonicity Theorem): For a normalized i -bit divisor $y_i = 1.b_1b_2\cdots b_{i-1}$, let $\text{recip}_a(y_i) = 0.1a_2a_3\cdots a_i a_{i+1}\cdots$ be a reciprocal approximation with relative error strictly less than $\frac{1}{2} 2^{-i}$ for all $y_i \in [1; 2 - 2^{-(i-1)})$. Then $\text{recip}(y_i) = \text{rn}_i(\text{recip}_a(y_i))$ is a monotonic, one-ulp reciprocal function over $[1; 2 - 2^{-(i-1)})$.

Proof: A reciprocal approximation $\text{recip}_a(y)$ with relative error strictly less than $\frac{1}{2} 2^{-i}$ satisfies $|\frac{1}{y_i} - \text{recip}_a(y_i)| < \frac{1}{2y_i} 2^{-i}$. So then $\text{rn}_i(\text{recip}_a(y_i))$ is a one-ulp reciprocal for $y_i \in [1; 2 - 2^{-(i-1)})$ satisfying $|\frac{1}{y_i} - \text{rn}_i(\text{recip}_a(y_i))| < (\frac{1}{2} + \frac{1}{2y_i}) 2^{-i}$. (Note that this bound scales down towards $\frac{3}{4}$ -ulp as $y \rightarrow 2$). For successive i -bit normalized divisors $y_i, y_i + 2^{-(i-1)} \in [1; 2]$, the difference of their reciprocals satisfies $\frac{1}{y_i} - \frac{1}{y_i + 2^{-(i-1)}} = \frac{2}{y_i(y_i + 2^{-(i-1)})} 2^{-i} \geq \frac{1}{y_i} 2^{-i}$. Assume that $\text{rn}_i(\text{recip}_a(y_i)) = \text{rd}_i(\frac{1}{y_i}) < \text{ru}_i(\frac{1}{y_i + 2^{-(i-1)}}) = \text{rn}_i(\text{recip}_a(y_i + 2^{-(i-1)}))$. Then the sum of these successive reciprocal rounding errors satisfies $(\frac{1}{y_i} - \text{rn}_i(\text{recip}_a(y_i)) + (\text{rn}_i(\text{recip}_a(y_i + 2^{-(i-1)})) - \frac{1}{y_i + 2^{-(i-1)}})) \geq (\frac{1}{y_i} - \frac{1}{y_i + 2^{-(i-1)}}) + 2^{-i} \geq (1 + \frac{1}{y_i}) 2^{-i}$. So at least one of these rounding errors is greater than or equal to $(\frac{1}{2} + \frac{1}{2y_i}) 2^{-i}$, a contradiction. Thus $\text{rn}_i(\text{recip}_a(y_i)) \geq \text{rn}_i(\text{recip}_a(y_i + 2^{-(i-1)}))$, and $\text{recip}(y_i) = \text{rn}_i(\text{recip}_a(y_i))$ is mono-

tonic for $y_i \in [1; 2 - 2^{-(i-1)}]$. \square

III. BIPARTITE TABLES

The bipartite table lookup process for determining an approximate reciprocal of a normalized binary divisor $y = 1.b_1b_2 \cdots b_{p-1}$, comprises the use of two distinct binary direct lookup tables of comparable size. These tables are concurrently addressed by distinct, equivalent length substrings of divisor bits, with each table fashioned to provide a distinct part of a carry save or borrow save representation of the approximate reciprocal. Specifically, our bipartite reciprocal approximations are of the form

$$\text{recip}(y) = \text{recip}_1(y_i) + \text{recip}_2(y)2^{-i}.$$

With $y_i = y_{2k} = 1.b_1b_2 \cdots b_{2k}$, the primary approximation $\text{recip}_1(y_{2k}) \in [\frac{1}{2}, 1]$ is determined by the $2k$ -bit index $b_1b_2 \cdots b_{2k}$. The secondary approximation term is determined by some leading bits and some supplementary trailing bits $b_{2k+1}b_{2k+2} \cdots$. The approximation may be fashioned so that $\text{recip}_2(y)$ is exclusively positive or negative, with magnitudes less than a unit, or sign-symmetric with magnitude less than half a unit. Partitioning the operand y into three equal k -bit parts, Figure 2 illustrates the look-up process, employing $3k$ leading bits of a higher precision divisor.

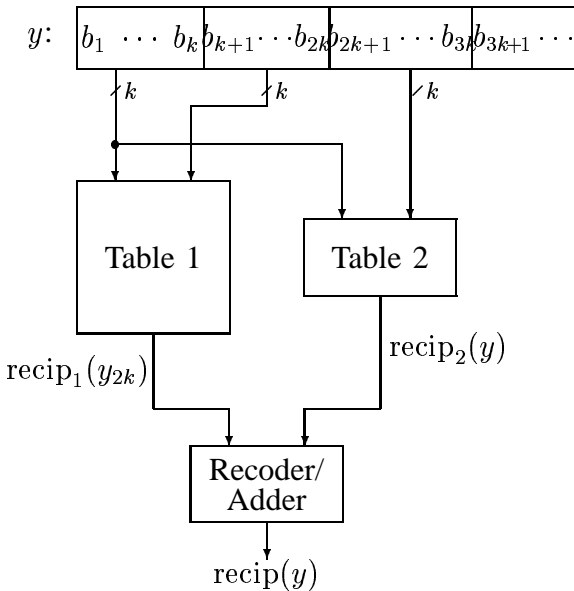


Fig. 2. The Bipartite Table Look-up Method

The compelling advantage of the two table bipartite lookup process is, that it provides a simple procedure to achieve essentially $1\frac{1}{2}$ times the

precision, at a cost of only twice the table size, compared to a single direct lookup table. For use as a seed or short reciprocal in application to division algorithms, the redundant reciprocal approximation may be sent directly to an appropriate multiplier recoder. For reciprocal function output in standard binary form the two outputs require a supplementary carry-completion addition.

Determination of the entries in a bipartite lookup table pair is guided by the following exact expansions particular to the reciprocal function:

Theorem 4 (Bipartite Reciprocal Identities): For the normalized binary divisor partition $y = y_i + f2^{-i}$ where $y_i = 1.b_1b_2 \cdots b_i$ and $f = 0.b_{i+1}b_{i+2} \cdots$ the reciprocal $\frac{1}{y} \in (\frac{1}{2}, 1]$ can be expanded to the sum of a primary term, determined by y_i , and a secondary term of magnitude less than 2^{-i} , according to any of the following

$$\frac{1}{y} = \frac{1}{y_i} - \frac{f}{yy_i}2^{-i} \quad (\text{borrow save expansion}) \quad (3)$$

$$\frac{1}{y} = \frac{1}{y_i + 2^{-i}} + \frac{1-f}{y(y_i + 2^{-i})}2^{-i} \quad (\text{carry save}) \quad (4)$$

$$\frac{1}{y} = \frac{1}{y_i + 2^{-(i+1)}} + \frac{\frac{1}{2} - f}{y(y_i + 2^{-(i+1)})}2^{-i} \quad (\text{midpoint}) \quad (5)$$

Proof: Putting the primary and secondary terms over a common denominator yields an immediate reduction verifying each identity. For borrow save $\frac{y}{yy_i} - \frac{f}{yy_i}2^{-i} = \frac{y-f2^{-i}}{yy_i} = \frac{y_i}{yy_i} = \frac{1}{y}$, and similarly for the carry save and midpoint expansion identities. \square

The claim that bipartite reciprocal approximations derived from (3) to (5) can have precision $\frac{3}{2}i$ bits is supported by the following observations. Let $i = 2k$ and consider the $\frac{3}{2}i = 3k$ input bits in Figure 2. A primary table employing the $i = 2k$ bit index $b_1b_2 \cdots b_{2k}$ with $\frac{3}{2}i = 3k$ bits of output, allows the primary term to be approximated with error less than half a unit in the $3k+1$ place. A secondary table uses index $b_1b_2 \cdots b_k \circ b_{2k+1}b_{2k+2} \cdots b_{3k}$, formed by concatenating k leading fraction bits of y with k leading bits of f . This table can provide a $(k+1)$ -bit output value for $\text{recip}_2(y)$, allowing the secondary term to be approximated to near the order of a unit in the $\frac{3}{2}i = 3k$ place. These arguments will now be made precise, leading to the specification of formulas for direct lookup table entries that minimize the maximum absolute errors in each of

the terms in expansions (3) to (5). For bipartite expansions it is most convenient to fix a common last place position for both terms, and minimize the maximum absolute error contributed by each term.

Note that the primary terms in each of the identities have exact inputs. Their evaluation can provide entries to i -bits-in j -bits-out direct lookup tables with an absolute error for each entry bounded by $2^{-(j+2)}$, due only to rounding the exact output to the output table size. E.g., for the borrow save expansion with $i = 2k$, let the output size be $3k + g$ bits where $g \geq 0$ is a small number of guard bits. Excluding the special case $y_i = 1$, the primary term approximation in the $2k$ -bits-in $(3k + g)$ -bits-out table is

$$\text{recip}_1(y_{2k}) = \text{rn}_{3k+g+1}\left(\frac{1}{y_{2k}}\right) = 0.1a_2a_3 \cdots a_{3k+g+1}.$$

The primary table size for $\text{recip}_1(y_{2k})$ is $2^{2k}(3k + g)$ bits with maximum absolute error $2^{-(3k+g+2)}$.

The secondary term approximation $\text{recip}_2(y) \approx -\frac{f}{yy_{2k}}$ for the borrow save reciprocal expansion (3) with $i = 2k$, will be determined from the k leading fraction bits of y , where $y_k = 1.b_1b_2 \cdots b_k$ along with the k leading bits of f with $f_k = .b_{2k+1}b_{2k+2} \cdots b_{3k}$. Thus we have $\text{recip}_2(y) = \text{recip}_2(y_k, f_k)$. In terms of the arguments (y_k, f_k) we note the following bounds on each of the factors of the secondary term $\frac{f}{yy_{2k}}$.

$$\begin{aligned} f_k &\leq f < f_k + 2^{-k} \\ y_k &\leq y_{2k} < y_k + 2^{-k} - 2^{-2k} \\ y_k + f_k 2^{-2k} &\leq y < y_k + 2^{-k} - 2^{-2k} + f_k 2^{-2k} + 2^{-3k} \end{aligned}$$

Lemma 5: For the divisor $y = 1.b_1b_2 \cdots$ with $2k \geq 4$, let $y_k = 1.b_1b_2 \cdots b_k$, $y_{2k} = 1.b_1b_2 \cdots b_{2k}$, $f = .b_{2k+1}b_{2k+2} \cdots$ and $f_k = .b_{2k+1}b_{2k+2} \cdots b_{3k}$. Then the borrow save expansion secondary term satisfies the following tight bounds which are tight in the sense that $\frac{f}{yy_{2k}}$ can be arbitrarily close to either bound:

$$\begin{aligned} &\frac{f_k}{(y_k + 2^{-k})(y_k + 2^{-k} - 2^{-2k} + f_k 2^{-2k} + 2^{-3k})} \\ &< \frac{f}{yy_{2k}} < \frac{f_k + 2^{-k}}{y_k(y_k + f_k 2^{-k})}. \end{aligned} \quad (6)$$

Let $m(y_k, f_k)$ be the midpoint of the interval determined by (6). The value $m(y_k, f_k)$ minimizes the maximum absolute error for approximation of

$\frac{f}{yy_{2k}}$ in each of the separate regions determined by each index $b_1b_2 \cdots b_k \circ b_{2k+1}b_{2k+2} \cdots b_k$. The maximum error over all the regions will occur for the argument pair $y_k = 1$, $f_k = 0$ with index $0^{[k]}$ leading to

Corollary 6: Let $m(y_k, f_k)$ be the midpoint of the interval determined by (6). Then

$$\left| \frac{f}{yy_{2k}} - m(y_k, f_k) \right| < \frac{3}{2} 2^{-3k}.$$

The secondary term in our bipartite borrow save approximation is then determined by rounding $m(y_k, f_k)$ to the last place position $k + g + 1$,

$$\text{recip}_2(y_k, f_k) = -\text{rn}_{k+g+1}(m(y_k, f_k)).$$

Including the two rounding errors we further obtain the following from Corollary 6.

Corollary 7: The borrow save bipartite reciprocal approximation for the normalized divisor $y = 1.b_1b_2 \cdots$ given by

$$\text{recip}(y) = \text{rn}_{3k+g+1}\left(\frac{1}{y_{2k}}\right) - \text{rn}_{k+g+1}(m(y_k, f_k)) 2^{-2k}$$

satisfies the absolute error bound

$$\left| \frac{1}{y} - \text{recip}(y) \right| < \left(\frac{3}{2} + \frac{1}{2^{g+1}} \right) 2^{-3k}.$$

For $g = 0$ the maximum error is then $2^{-(3k-1)}$ with total table size $2^{2k}(4k + 1)$. With just a few guard bits we approach the case for $g \rightarrow \infty$ where $\text{recip}(y) = \frac{1}{y_{2k}} - m(y_k, f_k) 2^{-2k}$ with $\left| \frac{1}{y} - \text{recip}(y) \right| < \frac{3}{4} 2^{-(3k-1)}$. It can be shown that the maximum relative error for such a bipartite table occurs for $y \rightarrow 1$, so the precision is at least $(3k - 1)$ -bits.

In practice bipartite tables are found most effective for total index lengths $9 \leq 3k \leq 18$, where each part has between 3 and 6 bits. Considering variable sized parts the preferred partitions of index parts are $k|k|k$, $(k + 1)|k|k$, and $(k + 1)|k|(k + 1)$.

Exploiting Symmetry in Bipartite Tables: The midpoint expansion (5) allows for design of a sign symmetric bipartite table process, providing one additional bit of accuracy. For the symmetric case some of the input bits and the secondary term approximation are subject to a conditional complementation.

When the approximate reciprocal type is a reciprocal function defined on 'exact' input points, the midpoint expansion (5) can be modified to yield a symmetric secondary term.

Symmetric Bipartite Reciprocal Functions: For the normalized p -bit divisor $y = 1.b_1b_2 \cdots b_{p-1}$, the secondary part of the partition $y = y_i + f2^{-i}$ has $f = 0.b_{i+1}b_{i+2} \cdots b_{p-1}$ with $f \in [0, 1 - 2^{-(p-i-1)}]$. The secondary part can be centered by subtracting $(\frac{1}{2} - 2^{-(p-i)})$ and adding the same to the primary part. The *symmetric divisor partition* for the p -bit normalized divisor $y = 1.b_1b_2 \cdots b_{p-1}$ is then

$$y = y_i - 2^{-p} + 2^{-(i+1)} + (f + 2^{-(p-i)} - \frac{1}{2})2^{-i}. \quad (7)$$

From (7) for any precision p the *symmetric bipartite identity* for $\frac{1}{y}$ is then

$$\frac{1}{y} = \frac{1}{y_i + 2^{-(i+1)} - 2^{-p}} + \frac{1 - 2(f + 2^{-(p-i)})}{y(y_i + 2^{-(i+1)} - 2^{-p})} 2^{-(i+1)}. \quad (8)$$

Then the *symmetric bipartite reciprocal function* for the $(3k+2)$ -bit normalized binary divisor $y = 1.b_1b_2 \cdots b_{3k-1}$ is determined from (8) with $i = 2k$ and $p = 3k+2$ by

$$\frac{1}{y} = \frac{1}{y_{2k} + 2^{-(2k+1)} - 2^{-p}} + \frac{(-1)^{b_{2k+1}} f^*}{y(y_{2k} + 2^{-(2k+1)} - 2^{-p})} 2^{-(2k+1)}$$

Here f^* is determined so that $1 - 2(f + 2^{-(k+2)}) = (-1)^{b_{2k+1}} f^*$ where

$$f^* = .b_{2k+2}^* b_{2k+3}^* \cdots b_{3k+1}^* 1 \\ = \begin{cases} .\bar{b}_{2k+2} \bar{b}_{2k+3} \cdots \bar{b}_{3k+1} 1 & \text{for } b_{2k+1} = 0 \\ .b_{2k+2} b_{2k+3} \cdots b_{3k+1} 1 & \text{for } b_{2k+1} = 1. \end{cases}$$

This allows the bounds

$$\frac{f^*}{(y_k + 2^{-k})^2} < \frac{f^*}{y(y_{2k} + 2^{-(2k+1)} - 2^{-p})} < \frac{f^*}{y_k^2} \quad (9)$$

where the interval midpoint $m^*(y_k, f^*)$ from (9)

is used to determine the second term of the symmetric bipartite approximate reciprocal function. The centering of the secondary part in (7) and (8) thus provides for a sharp result, since f^* is exact, and shares the practical convenience of determining f^* by a 1's complement.

IV. MULTIPARTITE TABLE LOOK-UP

The bipartite table lookup process for determining an approximate reciprocal can be expanded to a tripartite or multipartite process. The result is then the sum of three or more terms obtained from three or more table lookups indexed by comparably sized indices. Tripartite tables in principle should achieve

about $1\frac{3}{4}$ times the precision and cost about 3 times the table size as a single direct lookup table.

In practice multipartite tables are arguably most effective for tables with total input index lengths and resulting output approximation precisions in the range 15 to 24 bits. This range can be covered employing three to four-term sums with primary table indices bounded by eleven bits. These practical bounds keep total table size moderate. They also allow table lookup and subsequent addition time to be kept small.

For practical primary table indices of size at most 12 bits, the marginal improvement in tripartite and four-partite table approximations for each additional part is only 2-3 bits per part. For these index ranges the multipartite process is conveniently visualized by recognizing the divisor partition as a partial recoding operation.

Exploiting Recoding in Multipartite Tables:

Definition 8: Let $f = 0.b_1b_2 \cdots b_j1$ with $f \in [2^{-(j+1)}, 1 - 2^{-(j+1)}]$ and $j \geq 2k+3$. Then for $k \geq 1$, a k -digit *partial recoding* (Booth radix 4) of $(f - \frac{1}{2})$ denotes the expansion

$$f - \frac{1}{2} = d_1 2^{-2} + d_2 2^{-4} + d_3 2^{-6} + \cdots + d_k 2^{-2k} + t 2^{-(2k+1)}$$

with the *tail* t satisfying $t \in [-(1 - 2^{-(j-2k)}); (1 - 2^{-(j-2k)})]$ and $d_i \in \{-2, -1, 0, 1, 2\}$ for $1 \leq i \leq k$.

Note that the condition on the range of the tail $t \in [-(1 - 2^{-(j-2k)}); (1 - 2^{-(j-2k)})]$ for $f \in [2^{-(j+1)}; 1 - 2^{-(j+1)}]$ makes the expansion unique. In practice the digits d_i are determined from the bit triples $b_{2i-1}b_{2i}b_{2i+1}$ concurrently as in standard Booth recodings. The tail $t = .b_{2k+2}^* b_{2k+3}^* \cdots b_j^* 1$ is determined from conditionally complementing the bits $b_{2k+2}b_{2k+3} \cdots b_j$ depending on bit b_{2k+1} as described for symmetric bipartite expansions. The notion of partial recoding is extendable to Booth radix 8 recodings in the obvious way.

The divisor (input) partition for the bipartite midpoint expansion (5) is $y = (y_i + 2^{-i+1}) + (f - \frac{1}{2})2^{-i}$. This provides the basis for multipartite (output) expansions by partial recoding of the secondary term of (5).

Observation 9: Let the normalized binary divisor $y \in [1; 2)$ have the recoded tripartite partition $y = (y_i + 2^{-(i+1)}) + d 2^{-(i+2)} + t 2^{-(i+3)}$ with $y_i = 1.b_1b_2 \cdots b_i$, $d \in \{-2, -1, 0, 1, 2\}$, and $-1 \leq t < 1$.

Then

$$\frac{1}{y} = \frac{1}{y_i + 2^{-(i+1)}} - \frac{d}{y(y_i + 2^{-(i+1)})} 2^{-(i+2)} - \frac{t}{y(y_i + 2^{-(i+1)})} 2^{-(i+3)}. \quad (10)$$

Proof: The result is obtained by substituting the partial recoding $(f - \frac{1}{2}) = d2^{-2} + t2^{-3}$ into the bipartite midpoint expansion (5). \square

From (10) with $i = 2k$ we obtain a recoded tripartite expansion for use as a seed or short reciprocal,

$$\begin{aligned} \text{recip}(y) &= \text{rn}_{3k+g+2} \left(\frac{1}{y_{2k} + 2^{-(2k+1)}} \right) \\ &- d \left(\text{rd}_{k+g+1} \left(\frac{1}{(y_{2k} + 2^{-(2k+1)})^2} \right) + 2^{-(k+g+2)} \right) 2^{-(2k+2)} \\ &+ (-1)^{b_{2k+3}} \text{recip}_3(b_1 b_2 \cdots b_k \circ b_{2k+4}^* \cdots b_{3k+3}^*) 2^{-(2k+3)} \end{aligned}$$

The $2k$ -bit index $b_1 b_2 \cdots b_{2k}$ can retrieve both a $(3k+2+g)$ -bit output for $\text{rn}_{3k+2+g}(\frac{1}{y_{2k} + 2^{-(2k+1)}})$ and a $(k+1+g)$ -bit output for $\text{rn}_{k+g+1}(\frac{1}{(y_i + 2^{-(i+1)})^2})$. The second output can be conditionally complemented and/or shifted to determine $\text{recip}_2(y)$ as an approximation of $-\left(\frac{d}{y(y_i + 2^{-(i+1)})}\right)$ satisfying

$$\left| \text{recip}_2(y) - \left(-\frac{d}{y(y_i + 2^{-(i+1)})} \right) \right| 2^{-(2k+2)} < 2^{-(4k+2)} + 2^{-(3k+3+g)}.$$

The approximation for the terminal term $\text{recip}_3(y) \approx -\frac{t}{y(y_i + 2^{-(i+1)})}$ is handled as for the secondary term in the symmetric bipartite expansion, employing the $2k$ bit string $b_1 b_2 \cdots b_k \circ b_{2k+4}^* b_{2k+5}^* \cdots b_{3k+3}^*$ as the index to a separate terminal term table. The recoded tripartite expansion here employs an intermediate Booth 4 digit in the tripartite divisor partial recoding, to obtain a 2 bit enhancement of the precision of the result, compared to symmetric bipartite reciprocal approximation.

Analogous to Observation 9 we can employ a recoded 4-part divisor partition including two intermediate Booth radix 8 digits and obtain a 4-part identity

$$\frac{1}{y} = \frac{1}{y_i + 2^{-(i+1)}} - \frac{d_1}{y(y_i + 2^{-(i+1)})} 2^{-(i+3)} - \frac{d_0}{y(y_i + 2^{-(i+1)})} 2^{-(i+6)} - \frac{t}{y(y_i + 2^{-(i+1)})} 2^{-(i+7)}$$

The primary table can provide suitably rounded values for $\frac{1}{y_i + 2^{-(i+1)}}$, $\frac{1}{(y_i + 2^{-(i+1)})^2}$ and $\frac{3}{(y_i + 2^{-(i+1)})^2}$. The latter two values are sent to Booth radix 8 PPG's with input digits $-d_1$ and $-d_0$, providing the selected terms for $\text{recip}_2(y)$ and $\text{recip}_3(y)$. The final term is provided by a terminal term table with index $b_1 b_2 \cdots b_k \circ b_{2k+8}^* b_{2k+9}^* \cdots b_{3k+7}^*$ as previously described for the recoded tripartite expansion.

V. A SINGLE PRECISION, MONOTONIC ULP-ACCURATE RECIPROCAL FUNCTION

Let $y = 1.b_1 b_2 \cdots b_{23} \in [1; 2 - 2^{-23}]$. We split our reciprocal function into two cases, corresponding to two sub-intervals:

Case 1: $y \in [\frac{3}{2}; 2)$

Let y have the partition $y - y'_{11} + (f - \frac{1}{2})2^{-11}$ with sign-symmetric fractional part $f = .b_{12} b_{13} \cdots b_{-23} 1$ and $y'_{11} = 1.1 b_2 b_3 \cdots b_{11} + 2^{-12} - 2^{-24}$. Employing the symmetric bipartite identity $\frac{1}{y} = \frac{1}{y'_{11}} - \frac{1}{y} \frac{f - \frac{1}{2}}{y'_{11}}$ iteratively, we obtain

$$\frac{1}{y} = \frac{1}{y'_{11}} - \frac{f - \frac{1}{2}}{(y'_{11})^2} 2^{-11} + \frac{(2f - 1)^2}{y(y'_{11})^2} 2^{-24}.$$

Defining our constant term $c_0(b_2 b_3 \cdots b_{11})$ by adding half the maximum error

$$c_0 = \frac{1}{y'_{11}} + \frac{1 - 2^{-11}}{(y'_{11})^3} 2^{-25}$$

we obtain that $\frac{1}{y} = c_0 - \frac{f - \frac{1}{2}}{(y'_{11})^2} 2^{-11} + \frac{\delta}{y(y'_{11})^2} 2^{-25}$, where to a smaller order $|\delta| < 1$.

Let $f - \frac{1}{2}$ be partially recoded with two Booth radix 8 digits and a symmetric tail, then $f - \frac{1}{2} = d_1 2^{-3} + d_0 2^{-6} + t 2^{-7}$ with $d_0, d_1 \in \{-4, -3, \dots, 4\}$ and $|t| \leq \frac{63}{64}$, where

$$t = (-1)^{b_{18+1}} (.b_{19}^* b_{20}^* \cdots b_{23}^* 1).$$

Letting $y'_7 = 1.b_1 b_2 \cdots b_7 + 2^{-8} - 2^{-24}$, it can be shown that

$$\frac{t}{(y'_{11})^2} = \frac{t}{(y'_7)^2} + \frac{\gamma t}{y(y'_7)^2} 2^{-7}$$

with $|\gamma| < 1$. Then it can be shown that

$$\frac{1}{y} = c_0 - \frac{t}{(y'_7)^2} 2^{-18} - \frac{d_1}{(y'_{11})^2} 2^{-14} - \frac{d_0}{(y'_{11})^2} 2^{-17} + \frac{\alpha}{y(y'_{11})^2} 2^{-24} \quad (11)$$

with $|\alpha| < 1$. Since $b_1 = 1$ for $y \in [\frac{3}{2}; 2 - 2^{-23}]$, we can use a 10-bit index for determining simultaneously c_0 , $\frac{1}{(y'_{11})^2}$ and $\frac{3}{(y'_{11})^2}$, and another 11-bit index determines $\frac{t}{(y'_7)^2}$, all with sufficient guard bits. The first four terms of (11) then provide a four-partite reciprocal approximation to $\frac{1}{y}$, with error bound arbitrarily close to $\frac{1}{y(y'_{11})^2} 2^{-24}$.

Using 4 guard bits so that each of the four terms contributes a table based rounding error of at most 2^{-5} ulps, where here $\text{ulp} = 2^{-24}$, we obtain a four-partite reciprocal approximation $\text{recip}(y)$ satisfying $|\frac{1}{y} - \text{recip}_a(y)| < (\frac{1}{8} + \frac{1}{y(y'_{11})^2}) 2^{-24}$. Then $|\frac{1}{y} - \text{rn}_{24}(\text{recip}_a(y))| < (\frac{5}{8} + \frac{1}{y(y'_{11})^2}) 2^{-24}$.

Claim 1: $\text{rn}_{24}(\text{recip}_a(y))$ is a one-ulp monotonic reciprocal function over the interval $[\frac{3}{2}; 2 - 2^{-23}]$.

Claim 2: If $\text{rn}_{24}(\text{recip}_a(y))$ is not monotonic then at least one rounded reciprocal has an error at least $(\frac{1}{2} + \frac{1}{(y+2^{-24})^2})$ ulps.

Claim 2 can be verified by an argument similar to that of the proof of the Monotonicity Theorem (Theorem 3). Consider that the maximal total rounding error is essentially bounded by $(\frac{5}{8} + \frac{1}{y^3})$ ulps. Now $(\frac{1}{y^2} - \frac{1}{y^3}) > \frac{1}{8}$ over the interval $[\frac{3}{2}; 2 - 2^{-23}]$, so $(\frac{1}{2} + \frac{1}{y^2}) > \frac{5}{8} + \frac{1}{y^3}$. Therefore the error bound of $(\frac{5}{8} + \frac{1}{y^3})$ ulps is sufficient to guarantee that no error after rounding is as large as $(\frac{1}{2} + \frac{1}{y^2})$ ulps. It follows that $\text{rn}_{24}(\text{recip}_a(y))$ is monotonic for $y \in [\frac{3}{2}; 2 - 2^{-23}]$.

Since $(\frac{5}{8} + \frac{1}{y^3}) < 1$ also verifies a one-ulp bound for $y \in [\frac{3}{2}; 2 - 2^{-23}]$, the four-partite approximation $\text{rn}_{24}(\text{recip}_a(y))$, with $\text{recip}_a(y)$ given by the four terms in (11), is a single precision, one-ulp monotonic reciprocal function over $[\frac{3}{2}; 2 - 2^{-23}]$.

Figure 3 illustrates an implementation of this four-partite reciprocal function. Table 1 receives the 10-bit index $b_2 b_3 \dots b_{11}$ and outputs c_0 , $c_1 = -\frac{1}{(y'_{11})^2}$ and $3c_1$, with table values c_0 , $3c_1$ and $4c_1$ all rounded to position 2^{-28} .

The terms c_1 and $3c_1$ are each input to both multipliers, MG, where MG functions identically to a Booth radix-8 PPG. Table 2 receives the 11-bit index $b_2 \dots b_7 \circ b_{19}^* \dots b_{23}^*$ for determining $-\frac{t}{(y'_7)^2}$, rounded to position 2^{-28} . The sum is compressed

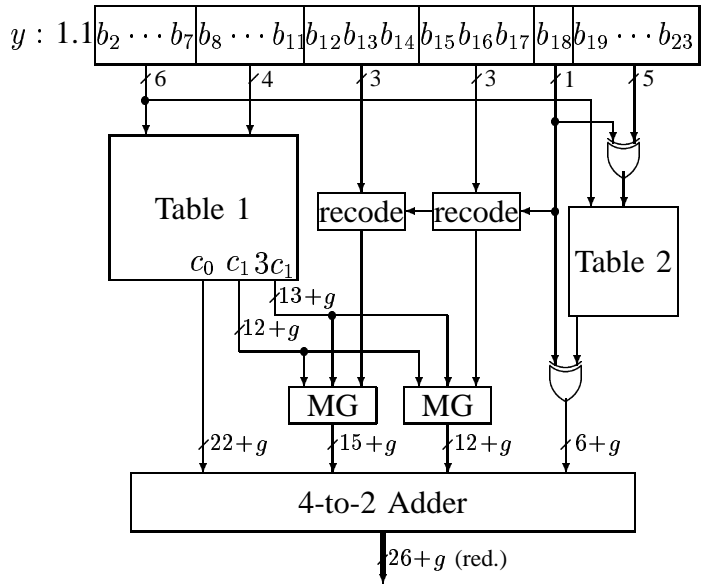


Fig. 3. Four-partite table reciprocal look-up for the interval $[\frac{3}{2}; 2]$.

in the 4-to-2 adder, maintaining guard bits, to a redundant reciprocal in the range $[\frac{1}{2}; \frac{2}{3}]$, including two leading guard digits [10].

For output as a single precision reciprocal the redundant result must be compressed by a carry-completing adder with rounding and normalization. For use as a divisor reciprocal, the result is recoded for multiplication by the single precision dividend, to obtain a quotient breakpoint by adaptively rounding with respect to the rounding mode (see Section I).

Case 2: $y \in [1; \frac{3}{2} - 2^{-23}]$

For this region we use 11 bits $(b_2 b_3 \dots b_{12})$ for the primary table index. Let y have the partition $y = y'_{12} + (f - \frac{1}{2})2^{-12}$ with $f = .b_{13} b_{14} \dots b_{23} 1$ and $y \mp \frac{f}{2^{12-1}} 2^{-12} + 2^{-13} - 2^{-24}$. Proceeding as in Case 1, the quadratic term is now $\frac{(2f-1)^2}{y(y'_{12})^2} 2^{-26}$, yielding an error term $\frac{\delta}{y(y'_{12})^2} 2^{-27}$ with $|\delta| < 1$, after centering by adjustment of $c_0(b_2 b_3 \dots b_{12})$. The terminal term now satisfies

$$\frac{t}{(y'_{12})^2} 2^{-19} = \frac{t}{(y'_8)^2} 2^{-19} + \frac{\gamma t}{y(y'_8)^2} 2^{-27},$$

where $\frac{t}{(y'_8)^2}$ still can be determined from an 11-bit index $b_2 b_3 \dots b_8 \circ b_{20}^* b_{21}^* b_{22}^* b_{23}^*$, since there is one less trailing bit, and one more leading bit than in Case 1.

We then obtain

$$\frac{1}{y} = c_0 - \frac{t}{(y'_8)^2} 2^{-19} - \frac{d_1}{(y'_{12})^2} 2^{-15} - \frac{d_0}{(y'_{12})^2} 2^{-18} + \frac{\alpha}{y(y'_{12})^2} 2^{56} \quad (12)$$

with $|\alpha| < 1$ for $y \in [1; \frac{3}{2} - 2^{-23}]$.

Then the four-partite approximation $\text{recip}_a(y)$, formed from the first four terms of (12) by rounding the table entries with four guard bits, will have a maximum error bound of $\frac{3}{8}$ ulps for $y \in [1; \frac{3}{2} - 2^{-23}]$. Then $\text{rn}_{24}(\text{recip}_a(y))$ is a one-ulp monotonic reciprocal function for $y \in [1; \frac{3}{2} - 2^{-23}]$.

Figure 4 illustrates the look-up table structure for implementing this reciprocal function over $[1; \frac{3}{2}]$. The tables of Figures 3 and 4 have combined size totalling less than 27 Kbytes, and the two structures can share much of the hardware shown, using suitably placed multiplexers.

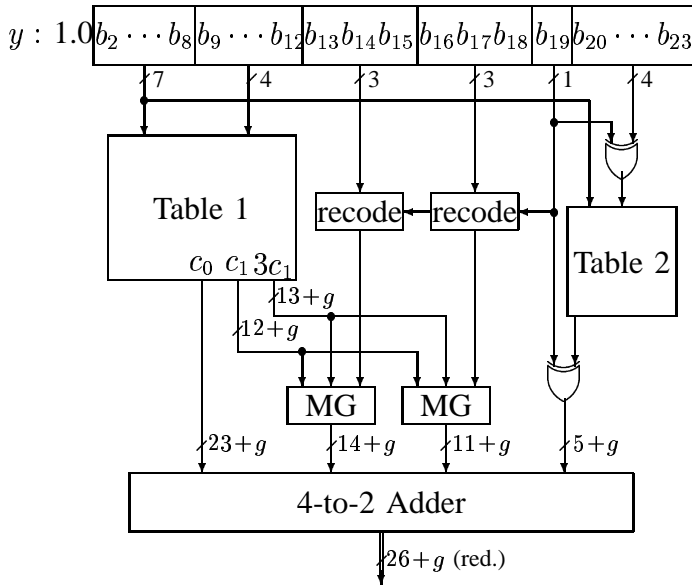


Fig. 4. Four-partite table reciprocal look-up for the interval $(1; \frac{3}{2})$.

REFERENCES

- [1] D. DasSarma and D. Matula, "Faithful Bipartite ROM Reciprocal Tables," in *Proc. 12th IEEE Symposium on Computer Arithmetic*. IEEE Computer Society, 1995, pp. 17–28.
- [2] H. Hassler and N. Takagi, "Function Evaluation by Table Look-Up and Addition," in *Proc. 12th IEEE Symposium on Computer Arithmetic*. IEEE, 1995, pp. 10–16.
- [3] M. Schulte and J. Stine, "Approximating Elementary Functions with Symmetric Bipartite Tables," *IEEE Transactions on Computers*, vol. 48, no. 8, pp. 842–847, 1999.

- [4] J.-M. Muller, "A Few Results on Table-Based Methods," *Reliable Computing*, vol. 5, no. 3, pp. 279–288, 1999.
- [5] C. Iordache and D. Matula, "Analysis of Reciprocal and Square Root Reciprocal Instructions in the AMD K6-2 Implementation of 3DNow," *Electronic Notes in Theoretical Computer Science*, vol. 24, 1999.
- [6] F. de Dinechin and A. Tisserand, "Some Improvements on Multipartite Table Methods," in *Proc. 15th IEEE Symposium on Computer Arithmetic*. IEEE, 2001, pp. 128–135.
- [7] W. Wong and E. Goto, "Fast Evaluation of the Elementary Functions in Single Precision," *IEEE Transactions on Computers*, vol. 44, no. 3, pp. 453–457, 1995.
- [8] J. Pineiro, J. Bruguera, and J.-M. Muller, "Faithful Powering Computation using Table Look-Up and a Fused Multiplication Tree," in *Proc. 15th IEEE Symposium on Computer Arithmetic*. IEEE, 2001, pp. 40–47.
- [9] F. de Dinechin and J. Detrey, "Multipartite Tables in JBits for the Evaluation of Functions on FPGA's," in *IEEE Reconfigurable Architecture Workshop, International Parallel and Distributed Symposium, Fort Lauderdale, Florida*. IEEE, April 2002.
- [10] P. Kornerup and J.-M. Muller, "Leading Guard Digits in Finite Precision Redundant Representations," 2004, submitted to ARITH17.