Contents

A Mechanically Checked Proof of IEEE Compliance of the Floating Point Multiplication, Division, and Square Root Algorithms of the AMD-K7^{TM} Processor

David M. Russinoff
Advanced Micro Devices, Inc.
Austin, TX

January 28, 1998

Abstract

We describe a mechanically verified proof of correctness of the floating point multiplication, division, and square root instructions of The AMD-K7 microprocessor. The instructions are implemented in hardware and represented here by register-transfer level specifications, the primitives of which are logical operations on bit vectors. On the other hand, the statements of correctness, derived from IEEE Standard 754, are arithmetic in nature and considerable more abstract. Therefore, we begin by developing a theory of bit vectors and their role in floating point representations and rounding. We then present the hardware model and a rigorous proof of its correctness. All of our definitions, lemmas, and theorems have been formally encoded in the ACL2 logic, and every step in the proof has been mechanically checked with the ACL2 prover.

1 Introduction

One of the challenges of formal hardware verification is the “semantic gap” between abstract behavioral specifications and concrete hardware models. Dealing effectively with this problem requires a formalism that is flexible enough to represent concepts at different levels of abstraction. In particular, specifications of floating point operations are most naturally expressed in numerical terms, while their hardware implementations are commonly modeled at the level of registers and bit vectors.

Conventional mathematical analysis may be usefully applied to numerical algorithms, but generally fails to provide any assurance regarding the correctness of hardware implementations. On the other hand, automatic finite-state techniques, which have been used to verify low-level specifications of arithmetic circuits [3, 4], lack the expressive power to represent high-level mathematical properties. General-purpose theorem provers offer an important alternative to finite-state tools, as they provide a framework for formal numerical analysis as well as mechanical support for checking properties of detailed low-level models.

In our previous work [8] and that of Moore et al. [6] on the AMD-K5 floating point unit, the ACL2 theorem prover [1] was used to support the verification of the IEEE compliance [5] of the AMD-K5 floating point division and square root operations. The implementation of these instructions was based on microcode that accessed existing hardware for addition, subtraction, multiplication, and rounding. It was appropriate, therefore, to model the instructions in a language in which the primitive operations included the computation of rounded products and sums, which were assumed to be implemented correctly. Consequently, the analysis was conveniently limited to the familiar realm of floating point numbers and rational arithmetic.

In contrast, the division and square root instructions of the AMD-K7 microprocessor, which were recently designed at AMD by Stuart Oberman [7], are implemented directly in hardware. In order to gain confidence in their correctness, it is desirable to model these instructions at the register-transfer level, where the basic operations are logical functions of bit vectors. Verification then requires bridging the gap between these low-level data and operations and the abstract mathematical objects and functions that they represent.

The subject of this paper is a mechanically verified proof of correctness of the AMD-K7 floating point multiplication, division, and square root instructions. The proof is based on a formal description of the hardware, derived from an executable model that was written in C and used for preliminary testing. The instructions are defined in terms of bitwise logical operations and integer addition and multiplication, which are treated as primitives.

The statements of correctness are based on IEEE standard 754 [5], which stipulates that each operation

> ... shall be performed as if it first produced an intermediate result correct to infinite precision and with unbounded range, and then rounded that result according to one of the [supported] modes ....

Thus, if $rnd(x, rc, pc)$ denotes the result of rounding a number x according to a specified rounding mode rc and degree of precision pc, and u is the value computed for the product of floating point numbers a and b in the context of rc and pc, then

$$u = rnd(a \cdot b, rc, pc). \tag{1}$$

Similarly, if v and w are the values computed for the quotient of a and b, and the square root of b, respectively, then

$$v = rnd(a/b, rc, pc) \tag{2}$$

and

$$w = rnd(\sqrt{b}, rc, pc). \tag{3}$$

The decision to use ACL2, however, has influenced our formulation of this last specification. As a subset of Common Lisp [9], ACL2 includes the rational numbers as a data type but not the reals. Consequently, we are somewhat limited in our formalization. The reader will notice that many of our lemmas are truths about real numbers but are presented here as propositions of rational arithmetic. More critically, since the square root itself is not a rational function, we are unable to formalize Equation (3) directly. Instead, we prove the following rational version: *For any nonnegative rational numbers ℓ and h, if $\ell^2 \leq P \leq h^2$, then*

$$rnd(\ell, rc, pc) \leq w \leq rnd(h, rc, pc). \tag{4}$$

As shown in [8], the equivalence of (3) and (4) is a simple consequence of (a) the monotonicity of rounding, and (b) the observation that for fixed rc and pc, the function rnd is constant in some neighborhood of any given irrational number.

Applied to the design of a device as complex as a floating point divider, mathematical proof provides a level of confidence that cannot be achieved through testing alone. In the present case, initial proof attempts revealed two design flaws that had survived some 80 million test vectors. The value of mechanical verification in this context is also clear: comprehensive analysis of a commercial floating point design is difficult if not impossible without computer assistance; in any case, the level of investment in its correctness requires a more efficient means of assurance than the conventional social process by which mathematical results are usually confirmed. This is not an argument, however, for circumventing the normal review process. The obligation to support a scientific claim cannot be satisfied simply by announcing that its correctness has been affirmed by an arcane automated proof system, the soundness of which itself is open to question. Moreover, the advantages of a coherent, surveyable proof extend beyond the issue of reliability: it is the only means by which a theory or result may be fully understood, applied, generalized, and assimilated into the mathematical domain. Traditional mathematical notation is clearly a better choice of medium for such an exposition than any formal language.

Since machine-assisted proofs have inherent advantages as well as disadvantages with respect to more traditional methods, we endeavor to combine the benefits of both approaches. In the following sections, we present a detailed proof of correctness, based on elementary mathematics and using only standard terminology and notation. In Section 2, we establish a general theory of floating point numbers, which should be reusable in a wide variety of applications. This is an extension of the theory presented in [8], including some additional properties of the rounding functions, but more significantly, a comprehensive treatment of bit vectors and their role in floating point representation. The specific hardware model is presented in Sections 3 and 4, along with precise formulations and detailed proofs of the above Equations (1), (2), and (4).

For the most part, each step in the proof may be readily checked by hand, requiring no special background in either mathematics or computer hardware. The only exception occurs in Section 4.2, where the accuracy of an approximation derived from a set of tables depends on properties of the tables that can only be verified by extensive (although straightforward) computation, involving approximately 10^5 table accesses and 10^6 arithmetic operations. The results of these calculations are stated without proof in Lemmas 4.1, 4.2, and 4.3.

On the other hand, along with the table calculations, every step in the proof, including every theorem and lemma presented below, has been formally encoded in the ACL2 logic and mechanically checked with the ACL2 prover, in the interest of eliminating the possibility of human error. The input to the prover, culminating in formal versions of our three main theorems, consisted of some 250 definitions and 3000 lemmas, in addition to the relevant definitions and lemmas of the previously developed general theory [8]. For the interested reader, the files containing this input are included as an appendix.

2 Floating Point Arithmetic

This section is a formalization of the floating point representation of rational numbers and rounding. The sets of rational numbers, positive rationals, integers, positive integers, and natural numbers (nonnegative integers) will be denoted by the symbols $\mathbb{Q}, \mathbb{Q}^+, \mathbb{Z}, \mathbb{Z}^+$, and $\mathbb{N}$, respectively. If $m \in \mathbb{Z}, n \in \mathbb{Z}^+$, and $m = nq + r$, where $q \in \mathbb{Z}, r \in \mathbb{N}$, and $r < n$, then we shall write $rem(m, n) = r$.

For $x \in \mathbb{Q}$, $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the *floor* and *ceiling* of x, respectively, defined to be the unique integers satisfying $\lfloor x \rfloor \leq x < \lfloor x \rfloor + 1$ and $\lceil x \rceil \geq x > \lceil x \rceil - 1$. We shall assume familiarity with the basic properties of these functions, including the following:

(1) If $n \in \mathbb{Z}$, then $\lfloor x + n \rfloor = \lfloor x \rfloor + n$.

(2) If $n \in \mathbb{Z}^+$, then $\lfloor \lfloor x \rfloor / n \rfloor = \lfloor x/n \rfloor$.

(3) If $m \in \mathbb{Z}$ and $n \in \mathbb{Z}^+$, then $\lfloor -(m+1)/n \rfloor = -\lfloor m/n \rfloor - 1$.

2.1 Bit Vectors

We shall exploit the natural correspondence between the bit vectors of length n and the natural numbers in the range $0 \leq x < 2^n$, under which the vector $b_{n-1}b_{n-2}\cdots b_1 b_0$, where each $b_k \in \{0,1\}$, corresponds to $x = \sum_{k=0}^{n-1} 2^k b_k$. The k^{th} bit of x, $x[k] = b_k$, is formally defined as follows:

Definition 2.1 *For all $x, k \in \mathbb{N}$, $x[k] = rem(\lfloor x/2^k \rfloor, 2)$.*

We have the following alternate characterization of $x[k]$:

Lemma 2.1 *For all $x, k \in \mathbb{N}$, $x[k] = \begin{cases} rem(x,2) & \text{if } k = 0 \\ \lfloor x/2 \rfloor [k-1] & \text{if } k > 0. \end{cases}$*

Proof: For $k > 0$, $x[k] = rem(\lfloor x/2^k \rfloor, 2) = rem(\lfloor \lfloor x/2 \rfloor / 2^{k-1} \rfloor, 2) = \lfloor x/2 \rfloor [k-1]$.□

Lemma 2.2 *For all $x, n, k \in \mathbb{N}$,*
(a) if $x < 2^n$, then $x[n] = 0$;
(b) if $k < n$ and $2^n - 2^k \leq x < 2^n$, then $x[k] = 1$.

Proof: (a) $x[n] = rem(\lfloor x/2^n \rfloor, 2) = rem(0,2) = 0$.
(b) Since $2^{n-k} - 1 \leq x/2^k < 2^{n-k}$, $rem(\lfloor x/2^k \rfloor, 2) = rem(2^{n-k} - 1, 2) = 1$. □

Lemma 2.3 *For all $x, m, n \in \mathbb{N}$,*
(a) $(x + 2^n)[n] \neq x[n]$; (b) if $m > n$, then $rem(x, 2^m)[n] = x[n]$.

Proof: For any $m \geq n$ and $q \in \mathbb{N}$,

$$(x + 2^m q)[n] = rem(\lfloor (x + 2^m q)/2^n \rfloor, 2) = rem(\lfloor x/2^n \rfloor + 2^{m-n} q, 2).$$

If $m = n$, then $rem(\lfloor x/2^n \rfloor + 2^{m-n}, 2) = rem(\lfloor x/2^n \rfloor + 1, 2) \neq rem(\lfloor x/2^n \rfloor, 2) = x[n]$; if $m > n$, then $2^{m-n}q$ is even and $rem(\lfloor x/2^n \rfloor + 2^{m-n}q, 2) = rem(\lfloor x/2^n \rfloor, 2) = x[n]$. □

The *left* and *right shift* functions take three arguments: a bit vector x, its length n, and a value $s \in \{0,1\}$ to be shifted in:

Definition 2.2 *Let $x, n, s \in \mathbb{N}$ with $x < 2^n$ and $s < 2$.*
(a) $shl(x, s, n) = rem(2x + s, 2^n)$; (b) $shr(x, s, n) = \lfloor x/2 \rfloor + 2^{n-1}s$.

Concatenation is also a function of three arguments: two bit vectors, x and y, and the length n of y:

Definition 2.3 *For all $x, y, n \in \mathbb{N}$, $cat(x, y, n) = 2^n x + y$.*

The following function extracts a field of bits:

Definition 2.4 *For all* $x, i, j \in \mathbb{N}$*,* $x[i:j] = \lfloor \mathit{rem}(x, 2^{i+1})/2^j \rfloor$*.*

Lemma 2.4 *For all* $x, y, i, j \in \mathbb{N}$*, if* $\mathit{rem}(x, 2^{i+1}) = \mathit{rem}(y, 2^{i+1})$*, then* $x[i:j] = y[i:j]$*.*

Proof: $x[i:j] = \lfloor \mathit{rem}(x, 2^{i+1})/2^j \rfloor = \lfloor \mathit{rem}(y, 2^{i+1})/2^j \rfloor = y[i:j]$. □

Lemma 2.5 *For all* $x, i, j, k, \ell \in \mathbb{N}$*,*
(a) if $i \geq k$ *and* $j \geq k$*, then* $x[i:j] = \lfloor x/2^k \rfloor [i-k : j-k]$*;*
(b) if $i \geq j+k$*, then* $x[i:j][k] = x[k+j]$*;*
(c) if $i \geq j+k$*, then* $x[i:j][k:\ell] = x[k+j : \ell+j]$*.*

Proof: (a) Let $x = 2^{i+1}q + r$, where $0 \leq r < 2^{i+1}$. Then

$$\lfloor x/2^k \rfloor = \lfloor 2^{i-k+1}q + r/2^k \rfloor = 2^{i-k+1}q + \lfloor r/2^k \rfloor,$$

hence

$$\mathit{rem}(\lfloor x/2^k \rfloor, 2^{i-k+1}) = \lfloor r/2^k \rfloor$$

and

$$\lfloor x/2^k \rfloor [i-k : j-k] = \lfloor \lfloor r/2^k \rfloor / 2^{j-k} \rfloor = \lfloor r/2^j \rfloor = \lfloor \mathit{rem}(x, 2^{i+1})/2^j \rfloor = x[i:j].$$

(b) Using Lemma 2.3,

$$\begin{aligned} x[i:j][k] &= \mathit{rem}(\lfloor \lfloor \mathit{rem}(x, 2^{i+1})/2^j \rfloor / 2^k \rfloor, 2) = \mathit{rem}(\lfloor \mathit{rem}(x, 2^{i+1})/2^{k+j} \rfloor, 2) \\ &= \mathit{rem}(x, 2^{i+1})[k+j] = x[k+j]. \end{aligned}$$

(c) Using (a),

$$\begin{aligned} x[i:j][k:\ell] &= \lfloor x/2^j \rfloor [i-j:0][k:\ell] = \mathit{rem}(\lfloor x/2^j \rfloor, 2^{i-j+1})[k:\ell] \\ &= \lfloor \mathit{rem}(\mathit{rem}(\lfloor x/2^j \rfloor, 2^{i-j+1}), 2^{k+1})/2^\ell \rfloor = \lfloor \mathit{rem}(\lfloor x/2^j \rfloor, 2^{k+1})/2^\ell \rfloor \\ &= \lfloor x/2^j \rfloor [k:\ell] = x[k+j : \ell+j]. \square \end{aligned}$$

We have two unary operations on bit vectors, *complement* and *decrement*:

Definition 2.5 *For all* $x, n \in \mathbb{N}$*, if* $x < 2^n$*, then*
(a) $\mathit{comp1}(x, n) = 2^n - x - 1$*;* *(b)* $\mathit{dec1}(x, n) = \mathit{rem}(2^n + x - 1, 2^n)$*.*

We have three binary logical operations, *and*, *or*, and *exclusive or*:

Definition 2.6 *For all* $x, y \in \mathbb{N}$*,*

$$(a) \quad x \mathbin{\&} y = \begin{cases} 0 & \textit{if } x = 0 \\ 2(\lfloor x/2 \rfloor \mathbin{\&} \lfloor y/2 \rfloor) + 1 & \textit{if } x \textit{ and } y \textit{ are both odd} \\ 2(\lfloor x/2 \rfloor \mathbin{\&} \lfloor y/2 \rfloor) & \textit{otherwise.} \end{cases}$$

$$(b) \quad x \mid y = \begin{cases} y & \textit{if } x = 0 \\ 2(\lfloor x/2 \rfloor \mid \lfloor y/2 \rfloor) & \textit{if } x \textit{ and } y \textit{ are both even} \\ 2(\lfloor x/2 \rfloor \mid \lfloor y/2 \rfloor) + 1 & \textit{otherwise.} \end{cases}$$

$$(c) \quad x \mathbin{\hat{}} y = \begin{cases} y & \textit{if } x = 0 \\ 2(\lfloor x/2 \rfloor \mathbin{\hat{}} \lfloor y/2 \rfloor) & \textit{if } \mathit{rem}(x, 2) = \mathit{rem}(y, 2) \\ 2(\lfloor x/2 \rfloor \mathbin{\hat{}} \lfloor y/2 \rfloor) + 1 & \textit{otherwise.} \end{cases}$$

The remainder of this subsection is a collection of properties of the binary logical operations.

Lemma 2.6 *For all $x, y \in \mathbb{N}$,*

(a) $x \mathbin{\&} y = 2(\lfloor x/2 \rfloor \mathbin{\&} \lfloor y/2 \rfloor) + (rem(x,2) \mathbin{\&} rem(y,2))$*;*
(b) $x \mathbin{|} y = 2(\lfloor x/2 \rfloor \mathbin{|} \lfloor y/2 \rfloor) + (rem(x,2) \mathbin{|} rem(y,2))$*.*

Proof: The equivalences are easily checked for all possible values of $rem(x,2)$ and $rem(y,2)$. □

Lemma 2.7 *For all $x, y, z \in \mathbb{N}$,*

(a) $x \mathbin{\&} 0 = 0$*;* *(e)* $(x \mathbin{\&} y) \mathbin{\&} z = x \mathbin{\&} (y \mathbin{\&} z)$*;*
(b) $x \mathbin{|} 0 = x$*;* *(f)* $(x \mathbin{|} y) \mathbin{|} z = x \mathbin{|} (y \mathbin{|} z)$*;*
(c) $x \mathbin{\&} y = y \mathbin{\&} x$*;* *(g)* $x \mathbin{|} (y \mathbin{\&} z) = (x \mathbin{|} y) \mathbin{\&} (x \mathbin{|} z)$*;*
(d) $x \mathbin{|} y = y \mathbin{|} x$*;* *(h)* $x \mathbin{\&} (y \mathbin{|} z) = (x \mathbin{\&} y) \mathbin{|} (x \mathbin{\&} z)$*.*

Proof: First note that Lemma 2.6 implies

$$\lfloor (x \mathbin{\&} y)/2 \rfloor = \lfloor x/2 \rfloor \mathbin{\&} \lfloor y/2 \rfloor \text{ and } rem(x \mathbin{\&} y, 2) = rem(x,2) \mathbin{\&} rem(y,2)$$

and

$$\lfloor (x \mathbin{|} y)/2 \rfloor = \lfloor x/2 \rfloor \mathbin{|} \lfloor y/2 \rfloor \text{ and } rem(x \mathbin{|} y, 2) = rem(x,2) \mathbin{|} rem(y,2).$$

We shall prove (h); the other proofs are similar:

It is easily verified that the statement holds for arguments in $\{0,1\}$. Thus,

$$\begin{aligned} rem(x \mathbin{\&} (y \mathbin{|} z), 2) &= rem(x,2) \mathbin{\&} rem(y \mathbin{|} z, 2) \\ &= rem(x,2) \mathbin{\&} (rem(y,2) \mathbin{|} rem(z,2)) \\ &= (rem(x,2) \mathbin{\&} rem(y,2)) \mathbin{|} (rem(x,2) \mathbin{\&} rem(z,2)) \\ &= rem(x \mathbin{\&} y, 2)) \mathbin{|} (rem(x \mathbin{\&} z, 2) \\ &= rem((x \mathbin{\&} y) \mathbin{|} (x \mathbin{\&} z), 2). \end{aligned}$$

Now, by inductive hypothesis,

$$\begin{aligned} \lfloor (x \mathbin{\&} (y \mathbin{|} z))/2 \rfloor &= \lfloor x/2 \rfloor \mathbin{\&} \lfloor (y \mathbin{|} z)/2 \rfloor \\ &= \lfloor x/2 \rfloor \mathbin{\&} (\lfloor y/2 \rfloor \mathbin{|} \lfloor z/2 \rfloor) \\ &= (\lfloor (x \mathbin{\&} y)/2 \rfloor) \mathbin{|} (\lfloor (x \mathbin{\&} z)/2 \rfloor) \\ &= (\lfloor x/2 \rfloor \mathbin{\&} \lfloor y/2 \rfloor) \mathbin{|} (\lfloor x/2 \rfloor \mathbin{\&} \lfloor z/2 \rfloor) \\ &= \lfloor ((x \mathbin{\&} y) \mathbin{|} (x \mathbin{\&} z))/2 \rfloor. \end{aligned}$$

Therefore,

$$\begin{aligned} x \mathbin{\&} (y \mathbin{|} z) &= \lfloor (x \mathbin{\&} (y \mathbin{|} z))/2 \rfloor + rem(x \mathbin{\&} (y \mathbin{|} z), 2) \\ &= \lfloor ((x \mathbin{\&} y) \mathbin{|} (x \mathbin{\&} z))/2 \rfloor + rem((x \mathbin{\&} y) \mathbin{|} (x \mathbin{\&} z), 2) \\ &= (x \mathbin{\&} y) \mathbin{|} (x \mathbin{\&} z). \Box \end{aligned}$$

Lemma 2.8 *Let $x, y, n \in \mathbb{N}$.*

(a) if $x < 2^n$ *and* $y < 2^n$*, then* $x \mathbin{|} y < 2^n$*;*
(b) if $y < 2^n$*, then* $(2^n x) \mathbin{|} y = 2^n x + y$*;*
(c) $(2^n x) \mathbin{|} (2^n y) = 2^n (x \mathbin{|} y)$*;*
(d) $rem(x \mathbin{|} y, 2^n) = rem(x, 2^n) \mathbin{|} rem(y, 2^n)$*.*

Proof: (a) For $n > 0$, $\lfloor x/2 \rfloor < 2^{n-1}$ and $\lfloor y/2 \rfloor < 2^{n-1}$, which implies $\lfloor x/2 \rfloor \mid \lfloor y/2 \rfloor < 2^{n-1}$, hence

$$x \mid y \leq 2(\lfloor x/2 \rfloor \mid \lfloor y/2 \rfloor) + 1 \leq 2(2^{n-1} - 1) + 1 < 2^n.$$

(b) For $n > 0$, since $\lfloor y/2 \rfloor < 2^{n-1}$,

$$\begin{aligned}
(2^n x) \mid y &= 2(\lfloor 2^n x/2 \rfloor \mid \lfloor y/2 \rfloor) + rem(2^n x, 2) \mid rem(y, 2) \\
&= 2(2^{n-1} x \mid \lfloor y/2 \rfloor) + 0 \mid rem(y, 2) \\
&= 2(2^{n-1} x + \lfloor y/2 \rfloor) + rem(y, 2) \\
&= 2^n x + 2\lfloor y/2 \rfloor + rem(y, 2) \\
&= 2^n x + y.
\end{aligned}$$

(c) For $n > 0$,

$$\begin{aligned}
(2^n x) \mid (2^n y) &= 2(\lfloor 2^n x/2 \rfloor \mid \lfloor 2^n x/2 \rfloor) + rem(2^n x, 2) \mid rem(2^n y, 2) \\
&= 2(2^{n-1} x \mid 2^{n-1} y) + 0 \mid 0 = 2(2^{n-1}(x \mid y)) + 0 \\
&= 2^n (x \mid y).
\end{aligned}$$

(d) Let $x = 2^n q_1 + r_1$ and $y = 2^n q_2 + r_2$, where $0 \leq r_1 < 2^n$ and $0 \leq r_2 < 2^n$. Then

$$\begin{aligned}
x \mid y &= (2^n q_1 + r_1) \mid (2^n q_2 + r_2) = (2^n q_1 \mid r_1) \mid (2^n q_2 \mid r_2) \\
&= (2^n q_1 \mid 2^n q_2) \mid (r_1 \mid r_2) = (2^n (q_1 \mid q_2)) \mid (r_1 \mid r_2) \\
&= 2^n (q_1 \mid q_2) + (r_1 \mid r_2).
\end{aligned}$$

But $r_1 \mid r_2 < 2^n$, hence $rem(x \mid y, 2^n) = r_1 \mid r_2 = rem(x, 2^n) \mid rem(y, 2^n)$. □

Lemma 2.9 *Let $x, y, n \in \mathbb{N}$.*

(a) $x \mathbin{\&} y \leq x$; *(c) $rem(x \mathbin{\&} y, 2^n) = rem(x, 2^n) \mathbin{\&} y$;*
(b) $2^n x \mathbin{\&} y = 2^n (x \mathbin{\&} \lfloor y/2^n \rfloor)$; *(d) if $x < 2^n$, then $x \mathbin{\&} y = x \mathbin{\&} rem(y, 2^n)$.*

Proof: (a) If $x = 0$, then $x \mathbin{\&} y = 0 \leq x$, and for $x > 0$,

$$\begin{aligned}
x \mathbin{\&} y &= 2(\lfloor x/2 \rfloor \mathbin{\&} \lfloor y/2 \rfloor) + (rem(x, 2) \mathbin{\&} rem(y, 2)) \leq 2\lfloor x/2 \rfloor + rem(x, 2) \\
&= x.
\end{aligned}$$

(b) For $n > 0$,

$$\begin{aligned}
2^n x \mathbin{\&} y &= 2(\lfloor 2^n x/2 \rfloor \mathbin{\&} \lfloor y/2 \rfloor) + rem(2^n x, 2) \mathbin{\&} rem(y, 2) \\
&= 2(2^{n-1} x \mathbin{\&} \lfloor y/2 \rfloor) + 0 \mathbin{\&} rem(y, 2) \\
&= 2(2^{n-1}(x \mathbin{\&} \lfloor \lfloor y/2 \rfloor / 2^{n-1} \rfloor)) + 0 \\
&= 2^n (x \mathbin{\&} \lfloor y/2^n \rfloor).
\end{aligned}$$

(c) Let $x = 2^n q + r$, $0 \leq r < 2^n$. Then $0 \leq r \mathbin{\&} y \leq r < 2^n$ and

$$\begin{aligned}
x \mathbin{\&} y &= (2^n q + r) \mathbin{\&} y = (2^n q \mid r) \mathbin{\&} y \\
&= (2^n q \mathbin{\&} y) \mid (r \mathbin{\&} y) = (2^n (q \mathbin{\&} \lfloor y/2^n \rfloor) \mid (r \mathbin{\&} y) \\
&= (2^n (q \mathbin{\&} \lfloor y/2^n \rfloor) + (r \mathbin{\&} y).
\end{aligned}$$

Therefore, $rem(x \mathbin{\&} y, 2^n) = r \mathbin{\&} y = rem(x, 2^n) \mathbin{\&} y$.

(d) Since $x \mathbin{\&} y \leq x < 2^n$, $x \mathbin{\&} y = rem(x \mathbin{\&} y, 2^n) = x \mathbin{\&} rem(y, 2^n)$. □

Lemma 2.10 *Let $x, y, n \in \mathbb{N}$.*
(a) $(x \mathbin{\&} y)[n] = x[n] \mathbin{\&} y[n]$; *(b)* $(x \mathbin{|} y)[n] = x[n] \mathbin{|} y[n]$.

Proof: The proofs are similar; we present the proof of (a), which proceeds by induction: For $n = 0$,

$$(x \mathbin{\&} y)[0] = \mathit{rem}(x \mathbin{\&} y, 2) = \mathit{rem}(x, 2) \mathbin{\&} \mathit{rem}(y, 2) = x[0] \mathbin{\&} y[0];$$

for $n > 0$,

$$\begin{aligned} (x \mathbin{\&} y)[n] &= \lfloor (x \mathbin{\&} y)/2 \rfloor [n-1] = (\lfloor x/2 \rfloor \mathbin{\&} \lfloor y/2 \rfloor)[n-1] \\ &= \lfloor x/2 \rfloor [n-1] \mathbin{\&} \lfloor y/2 \rfloor [n-1] = x[n] \mathbin{\&} y[n]. \square \end{aligned}$$

Lemma 2.11 *Let $x, n, k \in \mathbb{N}$, $k < n$.*
(a) $x \mathbin{\&} 2^k = 2^k x[k]$; *(c)* $x \mathbin{\&} (2^n - 2^k) = 2^k(x[n-1:k])$;
(b) $x \mathbin{|} 2^k = x + 2^k(1 - x[k])$;

Proof: (a) In the case $k = 0$, we have

$$x \mathbin{\&} 1 = 2(\lfloor x/2 \rfloor \mathbin{\&} 0) + \mathit{rem}(x, 2) = \mathit{rem}(x, 2) = x[0],$$

and for $k > 0$, by Lemma 2.1,

$$x \mathbin{\&} 2^k = 2(\lfloor x/2 \rfloor \mathbin{\&} 2^{k-1}) = 2(2^{k-1} \lfloor x/2 \rfloor [k-1]) = 2^k x[k].$$

(b) For $k = 0$, we have

$$x \mathbin{|} 1 = 2(\lfloor x/2 \rfloor \mathbin{|} 0) + 1 = 2\lfloor x/2 \rfloor + 1 = x + 1 - \mathit{rem}(x, 2) = x + 1 - x[0],$$

and for $k > 0$,

$$\begin{aligned} x \mathbin{|} 2^k &= 2\{\lfloor x/2 \rfloor \mathbin{|} 2^{k-1}\} + \mathit{rem}(x, 2) \\ &= 2\left\{\lfloor x/2 \rfloor + 2^{k-1}(1 - \lfloor x/2 \rfloor [k-1])\right\} + \mathit{rem}(x, 2) \\ &= 2\lfloor x/2 \rfloor + \mathit{rem}(x, 2) + 2^k(1 - \lfloor x/2 \rfloor [k-1]) \\ &= x + 2^k(1 - x[k]). \end{aligned}$$

(c) It suffices to prove the identity under the assumption $x < 2^n$, because then, by Lemmas 2.9 and 2.4, we have for arbitrary x:

$$x \mathbin{\&} (2^n - 2^k) = \mathit{rem}(x, 2^n) \mathbin{\&} (2^n - 2^k) = \mathit{rem}(x, 2^n)[n:k] = x[n:k].$$

For $k = 0$, we show by induction that $x \mathbin{\&} (2^n - 1) = x$. The case $n = 0$ is trivial, and for $n > 0$, since $\lfloor (2^n - 1)/2 \rfloor = 2^{n-1} - 1$, we have

$$\begin{aligned} x \mathbin{\&} (2^n - 1) &= 2(\lfloor x/2 \rfloor \mathbin{\&} (2^{n-1} - 1)) + \mathit{rem}(x, 2) \\ &= 2\lfloor x/2 \rfloor + \mathit{rem}(x, 2) = x. \end{aligned}$$

Now, for $k > 0$,

$$\begin{aligned} x \mathbin{\&} (2^n - 2^k) &= 2(\lfloor x/2 \rfloor \mathbin{\&} (2^{n-1} - 2^{k-1})) = 2 \cdot 2^{k-1} \lfloor x/2 \rfloor [n-2:k-1] \\ &= 2^k \lfloor \mathit{rem}(\lfloor x/2 \rfloor, 2^{n-1})/2^{k-1} \rfloor = 2^k \lfloor \lfloor x/2 \rfloor / 2^{k-1} \rfloor \\ &= 2^k \lfloor x/2^k \rfloor = 2^k(x[n-1:k]). \square \end{aligned}$$

Lemma 2.12 *Let $n, k, \ell \in \mathbb{N}$, $\ell \leq k < n$. Then*

$$(2^n - 2^\ell - 1) \,\&\, (2^n - 2^k) = \left\{ \begin{array}{ll} 2^n - 2^{k+1} & \textit{if } \ell = k \\ 2^n - 2^k & \textit{if } \ell < k. \end{array} \right.$$

Proof: Applying Lemma 2.11 (c), we have

$$\begin{aligned} (2^n - 2^\ell - 1) \,\&\, (2^n - 2^k) &= 2^k(2^n - 2^\ell - 1)[n-1:k] = 2^k\lfloor(2^n - 2^\ell - 1)/2^k\rfloor \\ &= 2^k(2^{n-k} + \lfloor -(2^\ell + 1)/2^k\rfloor \\ &= 2^n - 2^k(\lfloor 2^{\ell-k}\rfloor + 1).\Box \end{aligned}$$

2.2 Floating Point Representations

Floating point representation is based on the observation that every nonzero rational number x admits a unique factorization,

$$x = sgn(x)sig(x)2^{expo(x)},$$

where $sgn(x) \in \{1, -1\}$ (the *sign* of x), $1 \leq sig(x) < 2$ (the *significand* of x), and $expo(x) \in \mathbb{Z}$ (the *exponent* of x).

Definition 2.7 *Let $x \in \mathbb{Q}$. If $x \neq 0$, then*

(a) $sgn(x) = x/|x|$;
(b) $expo(x)$ is the unique integer that satisfies $2^{expo(x)} \leq |x| < 2^{expo(x)+1}$;
(c) $sig(x) = |x|2^{-expo(x)}$.

A floating point representation of x consists of three bit vectors, corresponding to $sgn(x)$, $sig(x)$, and $expo(x)$. A format is defined by the number of bits allocated to $sig(x)$ and $expo(x)$:

Definition 2.8 *Let $\phi = (\mu, \epsilon) \in \mathbb{Z}^+ \times \mathbb{Z}^+$. Then ϕ is a floating point format. A ϕ-***encoding** *is a triple $(s, m, e) \in \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ such that $s < 2$, $m < 2^\mu$, and $e < 2^\epsilon$. If $z = (s, m, e)$, then $s = $ get-sign(z), $m = $ get-man(z), and $e = $ get-expo(z). If $m \geq 2^{\mu-1}$, then z is a* **normal ϕ-encoding**.

The formats that are supported by the AMD-K7 floating point operations include $(24, 7)$, $(53, 10)$, and $(64, 15)$, which correspond to *single*, *double*, and *extended* precision as specified by IEEE, as well as a larger format, $(68, 18)$. In addition, in order to allow for the rounding error incurred by our iterative division and square root algorithms, which are required to produce results that are correctly rounded to 68 bits, the multiplier must support a somewhat more precise internal format. One of the objectives of our analysis is to determine the minimum required size of this format, and hence the minimum width of the multiplier. Thus, we introduce an integer parameter M, which represents the multiplier width and determines the internal format $(M, 18)$. We assume that $M \geq 75$, for as we shall see in Section 4, our proofs of correctness for division and square root will depend on this constraint.

In our formulation of the algorithms, the floating point formats are encoded as symbols:

Definition 2.9 *A* **precision control specifier** *is any of the symbols*

$$\texttt{PC-32}, \texttt{PC-64}, \texttt{PC-80}, \texttt{PC-87}, \textit{ and } \texttt{PC-*},$$

which correspond to the floating point formats

$$(24, 7), (53, 10), (64, 15), (68, 18), \textit{ and } (M, 18),$$

respectively. The first four of these symbols are called **external** *precision control specifiers. If π is any precision control specifier and $\phi = (\mu, \epsilon)$ is the corresponding format, then $mbits(\pi) = \mu$.*

The number x represented by a normal (μ, ϵ)-encoding (s, m, e) is given by $sgn(x) = (-1)^s$, $sig(x) = 2^{\mu-1}m$, and $expo(x) = e - (2^{\epsilon-1} - 1)$. Thus, the exponent is biased in order to provide for an exponent range $1 - 2^{\epsilon-1} \leq expo(x) \leq 2^{\epsilon-1}$:

Definition 2.10 *Let $z = (s, m, e)$ be a ϕ-encoding, where $\phi = (\mu, \epsilon)$ is a floating point format. Then $decode(z, \phi) = (-1)^s \cdot m \cdot 2^{e-2^{\epsilon-1}-\mu+2}$. In the case $\phi = (M, 18)$, we shall designate x simply as an* **encoding**, *and $decode(x, (M, 18))$ will be denoted as $\hat{x}$.*

Our characterization of the rational numbers that are represented by normal encodings is based on the following:

Definition 2.11 *Let $x \in \mathbb{Q}$ and $n \in \mathbb{Z}^+$. Then x is n-***exact** *iff $sig(x)2^{n-1} \in \mathbb{Z}$.*

The following basic property of n-exact numbers is proved in [8]:

Lemma 2.13 *If $x \in \mathbb{Q}^+$, $n \in \mathbb{Z}^+$, and x is n-exact, then the least n-exact number that is greater than x is $x + 2^{expo(x)+1-n}$.*

We shall also require this trivial characterization of n-exact bit vectors:

Lemma 2.14 *Let $x, n, k \in \mathbb{Z}^+$, $2^{n-1} \leq x < 2^n$. and $k < n$. The following are equivalent:*

(a) 2^k divides x; *(c) $x[n-1:k] = x/2^k$;*

(b) x is $(n-k)$-exact; *(d) $x[k-1:0] = 0$.*

Definition 2.12 *Let $x \in \mathbb{Q}$ and let $\phi = (\mu, \epsilon)$ be a floating point format. Then x is ϕ-***representable** *iff x is μ-exact and $-2^{\epsilon-1} + 1 \leq expo(x) \leq 2^{\epsilon-1}$. If $\phi = (M, 18)$, then we shall say that x is* **representable**.

The inverse of *decode* is given by the following:

Definition 2.13 *Let $\phi = (\mu, \epsilon)$ be a floating point format and let x be ϕ-representable, $x \neq 0$. Then $encode(x, \phi) = (s, m, e)$, where*

(a) if $sgn(x) = 1$, then $s = 0$, and if $sgn(x) = -1$, then $s = 1$;

(b) $m = sig(x)2^{\mu-1}$;

(c) $e = expo(x) + 2^{\epsilon-1} - 1$.

Lemma 2.15 *Let $\phi = (\mu, \epsilon)$ be a floating point format, let $z = (s, m, e)$ be a normal ϕ-encoding, and let $x = decode(z, \phi)$.*

(a) $sgn(x) = (-1)^s$; *(d) x is ϕ-representable;*

(b) $sig(x) = m/2^{\mu-1}$; *(e) $encode(x, \phi) = z$.*

(c) $expo(x) = e - 2^{\epsilon-1} + 1$;

Proof: Let $\phi = (\mu, \epsilon)$. Then

$$x = (-1)^s m 2^{e-(2^{\epsilon-1}-1)-\mu+1} = (-1)^s (m 2^{1-\mu}) 2^{e-(2^{\epsilon-1}-1)}.$$

But $2^{\mu-1} \leq m < 2^\mu$ yields $1 \leq m2^{1-\mu} < 2$, which implies (a), (b), and (c). Now (d) follows from the relation $0 \leq e < 2^\epsilon$, and (e) from Definition 2.13. □

2.3 Rounding

A *rounding mode* is a function $\mathcal{M}$ that computes an n-exact number $\mathcal{M}(x, n)$ corresponding to an arbitrary rational x and a degree of precision $n \in \mathbb{Z}^+$. We define five rounding modes:

Definition 2.14 *A* **rounding mode** *is any of the functions trunc, away, near, inf, and minf, where, for $x \in \mathbb{Q}$ and $n \in \mathbb{Z}^+$,*

(a) $trunc(x, n) = sgn(x)\lfloor 2^{n-1} sig(x) \rfloor 2^{expo(x)-n+1}$;

(b) $away(x, n) = sgn(x)\lceil 2^{n-1} sig(x) \rceil 2^{expo(x)-n+1}$;

(c) if $z = \lfloor 2^{n-1} sig(x) \rfloor$ *and* $f = 2^{n-1} sig(x) - z$, *then*

$$near(x, n) = \begin{cases} trunc(x, n) & \text{if } f < 1/2 \\ away(x, n) & \text{if } f > 1/2 \\ trunc(x, n) & \text{if } f = 1/2 \text{ and } z \text{ is even} \\ away(x, n) & \text{if } f = 1/2 \text{ and } z \text{ is odd}; \end{cases}$$

(d) $inf(x, n) = \begin{cases} away(x, n) & \text{if } x \geq 0 \\ trunc(x, n) & \text{if } x < 0; \end{cases}$

(e) $minf(x, n) = \begin{cases} trunc(x, n) & \text{if } x \geq 0 \\ away(x, n) & \text{if } x < 0. \end{cases}$

Only four of these modes are supported by the IEEE standard. In our representation of the algorithms, they will be encoded as symbols:

Definition 2.15 *A* **rounding control specifier** *is any of the symbols*

RC-CHOP, RC-POS, RC-NEG, *and* RC-NEAR,

which correspond to the rounding modes

trunc, inf, minf, and near,

respectively. Let ρ be a rounding control specifier corresponding to the rounding mode $\mathcal{M}$, let π be a precision control specifier, and let $x \in \mathbb{Q}$. Then

$$rnd(x, \rho, \pi) = \mathcal{M}(x, mbits(\pi)).$$

Some of the basic properties of the rounding modes, which are proved in [8], are listed in the following eight lemmas:

Lemma 2.16 *If $x \in \mathbb{Q}$, $\mathcal{M}$ is a rounding mode, and $n \in \mathbb{Z}^+$, then*

(a) $sgn(\mathcal{M}(x, n)) = sgn(x)$;
(b) if $\mathcal{M} \in \{trunc, away, near\}$, *then* $\mathcal{M}(-x, n) = -\mathcal{M}(x, n)$.

Lemma 2.17 *If $x, y \in \mathbb{Q}$, $x \leq y$, $\mathcal{M}$ is a rounding mode, and $n \in \mathbb{Z}^+$, then*

$$\mathcal{M}(x, n) \leq \mathcal{M}(y, n).$$

Lemma 2.18 *If $x \in \mathbb{Q}$, $\mathcal{M}$ is a rounding mode, and $n \in \mathbb{Z}^+$, then*
(a) $\mathcal{M}(x,n)$ is n-exact; *(b) if x is n-exact, then $x = \mathcal{M}(x,n)$.*

Lemma 2.19 *If $x \in \mathbb{Q}$, $\mathcal{M}$ is a rounding mode other than near, $m, n \in \mathbb{Z}^+$, and $m \leq n$, then*

$$\mathcal{M}(\mathcal{M}(x,n),m) = \mathcal{M}(x,m).$$

Lemma 2.20 *If $x \in \mathbb{Q}$ and $n \in \mathbb{Z}^+$, then*

$$|x| - 2^{expo(x)-n+1} < |trunc(x,n)| \leq |x| \leq |away(x,n)| < |x| + 2^{expo(x)-n+1}.$$

Lemma 2.21 *If $x \in \mathbb{Q}$ and $n \in \mathbb{Z}^+$, then*
(a) $expo(trunc(x,n)) = expo(x)$;
(b) $expo(away(x,n)) = expo(x)$ unless $|away(x,n)| = 2^{expo(x)+1}$.

Lemma 2.22 *If $x, a \in \mathbb{Q}$, $n \in \mathbb{Z}^+$, and a is n-exact, then*
(a) if $a \leq |x|$, then $a \leq |trunc(x,n)|$; *(b) if $a \geq |x|$, then $a \geq |away(x,n)|$.*

Lemma 2.23 *Let $x, y \in \mathbb{Q}$ and $n \in \mathbb{Z}^+$. If y is n-exact, then $|x - y| \geq |x - near(x,n)|$.*

We shall require a number of properties in addition to the above. The next lemma provides an implementation of truncation of bit vectors.

Lemma 2.24 *Let $x, m, n, k \in \mathbb{N}$. If $0 < k < n \leq m$ and $2^{n-1} \leq x < 2^n$, then*

$$trunc(x,k) = x \mathbin{\&} (2^m - 2^{n-k}).$$

Proof: By Lemma 2.11,

$$\begin{aligned} trunc(x,k) &= \lfloor 2^{k-1-expo(x)} x \rfloor 2^{expo(x)+1-k} = \lfloor x/2^{n-k} \rfloor 2^{n-k} \\ &= 2^{n-k}(x[n-1:n-k]) = x \mathbin{\&} (2^n - 2^{n-k}). \end{aligned}$$

But by Lemma 2.9,

$$x \mathbin{\&} (2^m - 2^{n-k}) = x \mathbin{\&} rem(2^m - 2^{n-k}, 2^n) = x \mathbin{\&} (2^n - 2^{n-k}).\Box$$

Lemma 2.24 is also the basis for our implementations of the other rounding modes, which therefore must be characterized in terms of truncation:

Lemma 2.25 *Let $x \in \mathbb{Q}^+$, $m \in \mathbb{Z}^+$, and $n \in \mathbb{Z}^+$. If x is m-exact and $m \geq n$, then*

$$away(x,n) = trunc(x + 2^{expo(x)+1}(2^{-n} - 2^{-m}), n).$$

Proof: Let $a = trunc(x + 2^{expo(x)+1}(2^{-n} - 2^{-m}), n)$. Since

$$a < x + 2^{expo(x)+1-n} \leq away(x,n) + 2^{expo(away(x,n))+1-n},$$

$a \leq away(x,n)$ by Lemma 2.13.

If x is n-exact, then $a \geq trunc(x,n) = x = away(x,n)$, and hence $a = away(x,n)$. Thus, we may assume x is not n-exact. But then since $x > trunc(x,n)$ and x is m-exact,

$$x \geq trunc(x,n) + 2^{expo(x)+1-m}$$

and hence

$$x + 2^{expo(x)+1}(2^{-n} - 2^{-m}) \geq trunc(x,n) + 2^{expo(x)+1-n} = away(x,n),$$

which implies $a \geq away(x,n)$. □

The remainder of this section addresses the properties of *near* rounding, concluding with its characterization as a truncated sum.

Lemma 2.26 *If $x \in \mathbb{Q}$ and $n \in \mathbb{Z}^+$, then $|x - near(x,n)| \leq 2^{expo(x)-n}$.*

Proof: By Lemma 2.16, we may assume $x > 0$. Let $a = trunc(x,n) + 2^{expo(x)+1-n}$. By Lemmas 2.18 and 2.23, if the statement fails, then

$$trunc(x,n) < x - 2^{expo(x)-n} < x + 2^{expo(x)-n} < away(x,n),$$

hence $a < away(x,n)$. Then by Lemmas 2.13 and 2.22(a), we have $a < x$, contradicting Lemma 2.22(b). □

Lemma 2.27 *Let $x \in \mathbb{Q}$ and $n \in \mathbb{Z}^+$. If x is $(n+1)$-exact but not n-exact, then*
(a) $trunc(x,n) = x - sgn(x)2^{expo(x)-n}$; *(b) $away(x,n) = x + sgn(x)2^{expo(x)-n}$.*

Proof Again we may assume $x > 0$. Let $a = x - 2^{expo(x)-n}$ and $b = x + 2^{expo(x)-n}$. Since $x > 2^{expo(x)}$, $x \geq 2^{expo(x)} + 2^{expo(x)+1-n}$ by Lemma 2.13, hence $a \geq 2^{expo(x)}$ and $expo(a) = expo(x)$.

By hypothesis, $x2^{n-expo(x)}$ is odd. Let $x2^{n-expo(x)} = 2k+1$. Then

$$a2^{n-1-expo(a)} = (x - 2^{expo(x)-n})2^{n-1-expo(x)} = (2k+1)/2 - 1/2 = k \in \mathbb{Z}.$$

Thus, a is n-exact, and by Lemma 2.13, so is $a + 2^{expo(a)+1-n} = b$. Now by Lemma 2.22, $a \leq trunc(x,n)$, but if $a < trunc(x,n)$, then Lemma 2.13 would imply $b \leq trunc(x,n)$, contradicting $x < b$. This establishes (a), and the proof of (b) is similar. □

Lemma 2.28 *Let $x, a \in \mathbb{Q}^+$, and $n \in \mathbb{Z}^+$. If a is n-exact, then*
(a) if $x > a + 2^{expo(a)-n}$, then $near(x,n) \geq a + 2^{expo(a)+1-n}$;
(b) if $x < a + 2^{expo(a)-n}$, then $near(x,n) \leq a$;
(c) if $x > a - 2^{expo(x)-n}$, then $near(x,n) \geq a$.

Proof: (a) Let $b = a + 2^{expo(a)+1-n}$. If $near(x,n) < b$, then Lemma 2.13 yields $near(x,n) \leq a$, hence $|near(x,n) - x| > |near(x,n) - b|$, contradicting Lemma 2.23.

(b) If $near(x,n) > a$, then $near(x,n) \geq b$, and a contradiction may be derived as in (a).

(c) By Lemma 2.17, we may assume $x < a$. Let $c = a - 2^{expo(x)+1-n}$. Then $c < x < a$. Since $a > x \geq 2^{expo(x)}$, $a \geq 2^{expo(x)} + 2^{expo(x)+1-n}$, and hence $x > c \geq 2^{expo(x)}$, which implies $expo(c) = expo(x)$. But $expo(c) \leq expo(a)$ and therefore

$$c2^{n-1-expo(c)} = a2^{n-1-expo(c)} - 1 \in \mathbb{Z},$$

i.e., c is n-exact. Now since $x > a - 2^{expo(x)-n} = c + 2^{expo(c)-n}$, (a) implies $near(x,n) \geq c + 2^{expo(c)+1-n} = a$. □

Lemma 2.29 *Let $n \in \mathbb{Z}$, $n > 1$, and $x \in \mathbb{Q}$. If x is $(n+1)$-exact but not n-exact, then $near(x,n)$ is $(n-1)$-exact.*

Proof: Again we may assume $x > 0$. Let $z = \lfloor 2^{n-1} sig(x) \rfloor$ and $f = 2^{n-1} sig(x) - z$. Since $2^{n-1} sig(x) \notin \mathbb{Z}$, $0 < f < 1$. But $2^n sig(x) = 2z + 2f \in \mathbb{Z}$, hence $2f \in \mathbb{Z}$ and $f = \frac{1}{2}$.

If z is even, then

$$near(x,n) = trunc(x,n) = z2^{expo(x)+1-n}$$

and by Lemma 2.21,

$$2^{n-2-expo(near(x,n))} near(x,n) = 2^{n-2-expo(x)} z2^{expo(x)+1-n} = z/2 \in \mathbb{Z}.$$

If z is odd, then

$$near(x,n) = away(x,n) = (z+1)2^{expo(x)+1-n}.$$

We may assume $away(x,n) \neq 2^{expo(x)+1}$, hence by Lemma 2.21,

$$2^{n-2-expo(near(x,n))} near(x,n) = 2^{n-2-expo(x)}(z+1)2^{expo(x)+1-n} = (z+1)/2 \in \mathbb{Z}.\Box$$

Lemma 2.30 *Let $n \in \mathbb{Z}$, $n > 1$, and $x \in \mathbb{Q}^+$. If $x + 2^{expo(x)-n} \geq 2^{expo(x)+1}$, then*

$$near(x,n) = 2^{expo(x)+1} = trunc(x + 2^{expo(x)-n}, n).$$

Proof: Suppose $near(x,n) \neq 2^{expo(x)+1}$. Then Lemma 2.21 implies $near(x,n) < 2^{expo(x)+1}$ and by Lemmas 2.13 and 2.26,

$$\begin{aligned} 2^{expo(x)+1} &\geq near(x,n) + 2^{expo(x)+1-n} \geq x - 2^{expo(x)-n} + 2^{expo(x)+1-n} \\ &= x + 2^{expo(x)-n} \geq 2^{expo(x)+1}. \end{aligned}$$

It follows that $x = 2^{expo(x)+1} - 2^{expo(x)-n}$ is $(n+1)$-exact but not n-exact, while $near(x,n) = 2^{expo(x)+1} - 2^{expo(x)+1-n}$ is n-exact but not $(n-1)$-exact, contradicting Lemma 2.29.

Now suppose $2^{expo(x)+1} \neq trunc(x + 2^{expo(x)-n}, n)$. Since $2^{expo(x)+1}$ is n-exact, $2^{expo(x)+1} < trunc(x + 2^{expo(x)-n}, n)$ by Lemma 2.22. But then by Lemma 2.13,

$$trunc(x + 2^{expo(x)-n}, n) \quad \geq \quad 2^{expo(x)+1} + 2^{expo(x)+2-n} > x + 2^{expo(x)-n}.\Box$$

Lemma 2.31 *If $n \in \mathbb{Z}$, $n > 1$, and $x \in \mathbb{Q}^+$, then*

$$near(x,n) = \begin{cases} trunc(x + 2^{expo(x)-n}, n-1) & \textit{if } x \textit{ is } (n+1)\textit{-exact but not } n\textit{-exact} \\ trunc(x + 2^{expo(x)-n}, n) & \textit{otherwise.} \end{cases}$$

Proof: If $x + 2^{expo(x)-n} \geq 2^{expo(x)+1}$, then by Lemmas 2.19 and 2.30,

$$near(x,n) = 2^{expo(x)+1} = trunc(x + 2^{expo(x)-n}, n) = trunc(x + 2^{expo(x)-n}, n-1).$$

Thus, we may assume $x + 2^{expo(x)-n} < 2^{expo(x)+1}$, and it follows from Lemmas 2.21 and 2.26 that

$$expo(near(x,n)) = expo(x + 2^{expo(x)-n}) = expo(x).$$

Case 1: x is n-exact

By Lemma 2.22, $trunc(x + 2^{expo(x)-n}, n) \geq x$. But since

$$trunc(x + 2^{expo(x)-n}, n) \leq x + 2^{expo(x)-n} < x + 2^{expo(x)+1-n},$$

Lemma 2.13 yields $trunc(x + 2^{expo(x)-n}, n) \leq x$, hence

$$trunc(x + 2^{expo(x)-n}, n) = x = near(x, n).$$

Case 2: x is not $(n+1)$-exact

We have $near(x,n) > x - 2^{expo(x)-n}$, for otherwise we would have $near(x,n) = x - 2^{expo(x)-n}$ by Lemma 2.26, and since $near(x,n)$ is $(n+1)$-exact, so would be

$$near(x,n) + 2^{expo(near(x,n))-n} = x - 2^{expo(x)-n} + 2^{expo(near(x,n))-n} = x.$$

Since $near(x,n) \leq x + 2^{expo(x)-n}$, $near(x,n) \leq trunc(x + 2^{expo(x)-n}, n)$ by Lemma 2.22. But since

$$trunc(x + 2^{expo(x)-n}, n) \leq x + 2^{expo(x)-n} < near(x,n) + 2^{expo(x)+1-n},$$

$trunc(x + 2^{expo(x)-n}, n) \leq near(x,n)$.

Case 3: x is $(n+1)$-exact but not n-exact

First suppose $near(x,n) > x$. Since $near(x,n)$ is $(n+1)$-exact, $near(x,n) \geq x + 2^{expo(x)-n}$, hence $near(x,n) = x + 2^{expo(x)-n}$, and by Lemma 2.29,

$$trunc(x + 2^{expo(x)-n}, n-1) = trunc(near(x,n), n-1) = near(x,n).$$

Now suppose $near(x,n) < x$. Then $near(x,n) < x + 2^{expo(x)-n}$ implies $near(x,n) \leq trunc(x + 2^{expo(x)-n}, n-1)$. But since

$$\begin{aligned} trunc(x + 2^{expo(x)-n}, n-1) &\leq x + 2^{expo(x)-n} = x - 2^{expo(x)-n} + 2^{expo(x)+1-n} \\ &< near(x,n) + 2^{expo(x)+2-n}, \end{aligned}$$

we have $trunc(x + 2^{expo(x)-n}, n-1) \leq near(x,n)$. □

3 Multiplication

3.1 The Program *FPU-MUL*

The multiplication algorithm is represented by the program *FPU-MUL*, as listed in Figures 1 and 2. The program is coded in a simple language, consisting of assignments and conditional branches. The primitive operations are logical operations on bit vectors and integer addition and multiplication, the implementation of which is not addressed here.

The algorithm is intended to be implemented with three distinct (integer) multipliers, which operate on the same two M-bit factors, yielding identical products of either $2M$ or $2M-1$ bits. The output of the first multiplier is manipulated under the assumption that *overflow* occurs, i.e., the product has $2M$ bits. In parallel, the output of the second multiplier is similarly manipulated under the opposite assumption. Meanwhile, the most significant bit produced by the third multiplier is examined to determine which of the first two results will actually be used while the other is discarded.

The inputs to this program include two encodings, x and y, of the numbers to be multiplied, as well as two specifiers, rc and pc, which control the rounding of the product. Irrespective of this rounding, the result is returned in the $(M, 18)$ format. Thus, the output z is expected to satisfy

$$\hat{z} = rnd(\hat{x}\hat{y}, rc, pc).$$

```
Program FPU-MUL(op,pc,lastpc,rc,x,y,z,r,d,inexact):

$sign \leftarrow get\text{-}sign(x)$ ^ $get\text{-}sign(y)$;
$man\text{-}unrounded \leftarrow get\text{-}man(x) \cdot get\text{-}man(y)$;
$overflow \leftarrow man\text{-}unrounded[2M-1]$;
if $man\text{-}unrounded[lsb(pc)-3:0] = 0$
   then $sticky\text{-}no\text{-}overflow \leftarrow 0$
   else $sticky\text{-}no\text{-}overflow \leftarrow 1$;
$sticky\text{-}with\text{-}overflow \leftarrow sticky\text{-}no\text{-}overflow \mid man\text{-}unrounded[lsb(pc)-2]$;
$inexact\text{-}no\text{-}overflow \leftarrow sticky\text{-}with\text{-}overflow$;
$inexact\text{-}with\text{-}overflow \leftarrow inexact\text{-}no\text{-}overflow \mid man\text{-}unrounded[lsb(pc)-1]$;
if $op$ = OP-BACK
   then if $overflow = 1$
            then $inexact \leftarrow inexact\text{-}with\text{-}overflow$
            else $inexact \leftarrow inexact\text{-}no\text{-}overflow$;
if $op$ = OP-BACK then
      $rconst\text{-}with\text{-}overflow \leftarrow comp1(2^M get\text{-}man(d), 2M)$
   else if $op$ = OP-LAST then
      $rconst\text{-}with\text{-}overflow \leftarrow 2^{lsb(lastpc)-2}$
   else if $rc$ = RC-NEAR then
      $rconst\text{-}with\text{-}overflow \leftarrow 2^{lsb(pc)-1}$
   else if ($sign = 1 \wedge rc$ = RC-NEG) $\vee$ ($sign = 0 \wedge rc$ = RC-POS) then
      $rconst\text{-}with\text{-}overflow \leftarrow 2^{lsb(pc)} - 1$
   else $rconst\text{-}with\text{-}overflow \leftarrow 0$;
$rconst\text{-}no\text{-}overflow \leftarrow shr(rconst\text{-}with\text{-}overflow, 0, 2M)$;
if $op$ = OP-BACK
   then {$add\text{-}with\text{-}overflow \leftarrow (man\text{-}unrounded + rconst\text{-}with\text{-}overflow + 1)[2M:0]$;
         $add\text{-}no\text{-}overflow \leftarrow (man\text{-}unrounded + rconst\text{-}no\text{-}overflow + 1)[2M-1:0]$}
   else {$add\text{-}with\text{-}overflow \leftarrow (man\text{-}unrounded + rconst\text{-}with\text{-}overflow)[2M:0]$;
         $add\text{-}no\text{-}overflow \leftarrow (man\text{-}unrounded + rconst\text{-}no\text{-}overflow)[2M-1:0]$};
$round\text{-}carryout\text{-}no\text{-}overflow \leftarrow add\text{-}no\text{-}overflow[2M-1]$;
$round\text{-}carryout\text{-}with\text{-}overflow \leftarrow add\text{-}with\text{-}overflow[2M]$;
if $op$ = OP-LAST
   then {$trunc\text{-}with\text{-}overflow \leftarrow 2^{2M} - 2^{lsb(lastpc)-1}$;
         $trunc\text{-}no\text{-}overflow \leftarrow 2^{2M} - 2^{lsb(lastpc)-2}$}
   else {$trunc\text{-}with\text{-}overflow \leftarrow 2^{2M} - 2^{lsb(pc)}$;
         $trunc\text{-}no\text{-}overflow \leftarrow 2^{2M} - 2^{lsb(pc)-1}$};
```

Figure 1: *FPU-MUL*

```
if $rc = \texttt{RC-NEAR} \wedge \mathit{sticky\text{-}no\text{-}overflow} = 0 \wedge \mathit{add\text{-}no\text{-}overflow}[\mathit{lsb}(\mathit{pc}) - 2] = 0$
  then $\mathit{man\text{-}rounded\text{-}no\text{-}overflow}$
         $\leftarrow (2^{2M-2}\mathit{round\text{-}carryout\text{-}no\text{-}overflow} \mid \mathit{add\text{-}no\text{-}overflow})$
            $\texttt{\&}\ ((2^{2M} - 1 - 2^{\mathit{lsb}(\mathit{pc})-1})\ \texttt{\&}\ \mathit{trunc\text{-}no\text{-}overflow})$
  else $\mathit{man\text{-}rounded\text{-}no\text{-}overflow}$
         $\leftarrow (2^{2M-2}\mathit{round\text{-}carryout\text{-}no\text{-}overflow} \mid \mathit{add\text{-}no\text{-}overflow})$
            $\texttt{\&}\ \mathit{trunc\text{-}no\text{-}overflow}$;
if $rc = \texttt{RC-NEAR} \wedge \mathit{sticky\text{-}with\text{-}overflow} = 0 \wedge \mathit{add\text{-}with\text{-}overflow}[\mathit{lsb}(\mathit{pc}) - 1] = 0$
  then $\mathit{man\text{-}rounded\text{-}with\text{-}overflow}$
         $\leftarrow (2^{2M-1}\mathit{round\text{-}carryout\text{-}with\text{-}overflow} \mid \mathit{add\text{-}with\text{-}overflow})$
            $\texttt{\&}\ ((2^{2M} - 1 - 2^{\mathit{lsb}(\mathit{pc})})\ \texttt{\&}\ \mathit{trunc\text{-}with\text{-}overflow})$;
  else $\mathit{man\text{-}rounded\text{-}with\text{-}overflow}$
         $\leftarrow (2^{2M-1}\mathit{round\text{-}carryout\text{-}with\text{-}overflow} \mid \mathit{add\text{-}with\text{-}overflow})$
            $\texttt{\&}\ \mathit{trunc\text{-}with\text{-}overflow}$;
$\mathit{exp\text{-}unrounded} \leftarrow (\mathit{get\text{-}expo}(x) + \mathit{get\text{-}expo}(y) + 2^{17} + 1)[17:0]$;
$\mathit{exp\text{-}rounded\text{-}with\text{-}overflow}$
  $\leftarrow (\mathit{exp\text{-}unrounded} + \mathit{round\text{-}carryout\text{-}with\text{-}overflow} + 1)[17:0]$;
$\mathit{exp\text{-}rounded\text{-}no\text{-}overflow} \leftarrow (\mathit{exp\text{-}unrounded} + \mathit{round\text{-}carryout\text{-}no\text{-}overflow})[17:0]$;
if $\mathit{get\text{-}man}(x) = 0$ then
    $z \leftarrow (\mathit{sign}, 0, \mathit{get\text{-}expo}(x))$
  else if $\mathit{get\text{-}man}(y) = 0$ then
    $z \leftarrow (\mathit{sign}, 0, \mathit{get\text{-}expo}(y))$
  else if $\mathit{overflow} = 1$ then
    $z \leftarrow (\mathit{sign}, \mathit{man\text{-}rounded\text{-}with\text{-}overflow}[2M-1:M], \mathit{exp\text{-}rounded\text{-}with\text{-}overflow})$
  else $z \leftarrow (\mathit{sign}, \mathit{man\text{-}rounded\text{-}no\text{-}overflow}[2M-2:M-1], \mathit{exp\text{-}rounded\text{-}no\text{-}overflow})$;
if $\mathit{op} = \texttt{OP-DIV}$ then
  if $\mathit{overflow} = 1$ then
    $r \leftarrow (0, \mathit{comp1}(\mathit{man\text{-}unrounded}, 2M)[2M-2:M-1], 2^{17} - 2)$
  else if $\mathit{round\text{-}carryout\text{-}no\text{-}overflow} = 0$ then
    $r \leftarrow (0, \mathit{comp1}(\mathit{man\text{-}unrounded}, 2M)[2M-1:M], 2^{17} - 1)$
  else $r \leftarrow (0, 2^M - 1, 2^{17} - 2)$
else if $\mathit{op} = \texttt{OP-SQRT}$ then
  if $\mathit{overflow} = 1$ then
    $r \leftarrow (0, (\mathit{comp1}(\mathit{man\text{-}unrounded}, 2M) \mid 2^{2M-1})[2M-1:M], 2^{17} - 2)$
  else if $\mathit{round\text{-}carryout\text{-}no\text{-}overflow} = 0$ then
    $r \leftarrow (0, \mathit{shr}(\mathit{comp1}(\mathit{man\text{-}unrounded}, 2M)[2M-2:0], 1, 2M)[2M-1:M], 2^{17} - 1)$
  else $r \leftarrow (0, 2^M - 1, 2^{17} - 2)$
```

Figure 2: *FPU-MUL (continued)*

As a notational convenience, the following function gives the position of the least significant bit of a $2M$-bit integer that has been rounded to a given degree of precision:

Definition 3.1 *For any precision control specifier π, $lsb(\pi) = 2M - mbits(\pi)$.*

In addition to computing products, the multiplication hardware performs several auxiliary functions in support of the divide and square root operations. These are specified by the input *op*, the value of which may be any of the symbols `OP-MUL`, `OP-DIV`, `OP-SQRT`, `OP-LAST`, and `OP-BACK`.

Basic floating point multiplication is performed in the case $op =$ `OP-MUL`: the inputs x and y are simply multiplied and rounded according to the specifiers *pc* and *rc*, and the IEEE compliant result is returned as the output z, as described by Theorem 1. The same holds for $op =$ `OP-DIV` and $op =$ `OP-SQRT`, but an additional output r is returned in these cases: for `OP-DIV`, $\hat{r}$ is an approximation of $2 - \hat{x}\hat{y}$; for `OP-SQRT`, $\hat{r}$ is an approximation of $(3 - \hat{x}\hat{y})/2$. The errors of these approximations are given by Lemma 3.5.

When *FPU-MUL* is called by division or square root, *pc* is always `PC-*`, indicating the internal format $(M, 18)$. However, on the final iteration of either of these operations, signaled by `OP-LAST`, the product is rounded to a lower precision, as determined by the input *lastpc*. This behavior is described formally by Lemma 3.7.

Finally, the symbol `OP-BACK` indicates a *back multiplication* to determine whether the product previously computed by `OP-LAST` is an overestimate or an underestimate of the exact value sought. The value given by the input d is subtracted from the product of x and y. In the case of division, x is the denominator, y is the approximate quotient, and d is the numerator; in the square root case, both x and y are the approximate square root and d is the radicand. In both cases, the results of the comparison are given by the outputs z and *inexact*, as stated in Lemma 3.8.

Thus, our analysis will be based on an execution of

$$\mathit{FPU\text{-}MUL(op,pc,lastpc,rc,x,y,z,r,d,inexact)},$$

under the following assumptions regarding the inputs:

(a) $op \in \{$`OP-MUL`, `OP-DIV`, `OP-SQRT`, `OP-LAST`, `OP-BACK`$\}$;

(b) *pc* is a precision control specifier;

(c) if $op =$ `OP-LAST`, then *lastpc* is an external precision control specifier;

(d) *rc* is a rounding control specifier;

(e) x and y are normal encodings;

(f) if $op =$ `OP-BACK`, then d is a normal encoding.

3.2 Basic Results

For convenience, we introduce several auxiliary variables. First, we define

$$sticky = \begin{cases} sticky\text{-}with\text{-}overflow & \text{if } overflow = 1 \\ sticky\text{-}no\text{-}overflow & \text{if } overflow = 0. \end{cases}$$

Each of the variables *rconst*, *add*, *round-carryout*, *trunc*, *man-rounded*, and *expo-rounded* is defined in the analogous manner. We also define

$$P = \begin{cases} 2M & \text{if } overflow = 1 \\ 2M-1 & \text{if } overflow = 0, \end{cases}$$

$$\mu = mbits(pc),$$

and

$$trunc' = \begin{cases} trunc, & \text{if } rc \neq \texttt{RC-NEAR} \text{ or } sticky = 1 \text{ or } add[P-\mu-1] = 1 \\ trunc \,\&\, (2^{2M} - 1 - 2^{P-\mu}), & \text{otherwise.} \end{cases}$$

Lemma 3.1
(a) $sig(man\text{-}unrounded) = sig(\hat{x})sig(\hat{y})/2^{overflow}$;
(b) $expo(man\text{-}unrounded) = P - 1$;
(c) $sig(\hat{x}\hat{y}) = sig(man\text{-}unrounded)$;
(d) $expo(\hat{x}\hat{y}) = expo(\hat{x}) + expo(\hat{y}) + overflow$.

Proof: Since x and y are normal encodings,

$$2^{2M-2} \leq man\text{-}unrounded = get\text{-}man(x) \cdot get\text{-}man(y) < 2^{2M},$$

and (b) follows from Lemma 2.2.

By Lemma 2.15,

$$\begin{aligned} man\text{-}unrounded &= 2^{M-1}sig(\hat{x})2^{M-1}sig(\hat{y}) \\ &= sig(\hat{x})sig(\hat{y})2^{-overflow}2^{2M-2+overflow} \\ &= sig(\hat{x})sig(\hat{y})2^{-overflow}2^{expo(man\text{-}unrounded)}, \end{aligned}$$

which implies (a).

To derive (c) and (d), we need only observe that

$$\begin{aligned} \hat{x}\hat{y} &= sgn(\hat{x})sig(\hat{x})2^{expo(\hat{x})}sgn(\hat{y})sig(\hat{y})2^{expo(\hat{y})} \\ &= sgn(\hat{x}\hat{y})\left[sig(\hat{x})sig(\hat{y})/2^{overflow}\right]2^{expo(\hat{x})+expo(\hat{y})+overflow}.\Box \end{aligned}$$

Lemma 3.2
(a) $sticky = 0$ *iff* $man\text{-}unrounded$ *is* $(\mu+1)$*-exact;*
(b) $inexact = 0$ *iff* $man\text{-}unrounded$ *is* μ*-exact;*

Proof: We have $sticky\text{-}no\text{-}overflow = 0 \Leftrightarrow 2^{lsb(pc)-2}$ divides $man\text{-}unrounded$, and

$$\begin{aligned} & sticky\text{-}with\text{-}overflow = 0 \\ & \Leftrightarrow 2^{lsb(pc)-2} \text{ divides } man\text{-}unrounded \text{ and } man\text{-}unrounded[lsb(pc)-2] = 0 \\ & \Leftrightarrow 2^{lsb(pc)-2} \text{ divides } man\text{-}unrounded \text{ and 2 divides } man\text{-}unrounded/2^{lsb(pc)-2} \\ & \Leftrightarrow 2^{lsb(pc)-1} \text{ divides } man\text{-}unrounded. \end{aligned}$$

Thus, $sticky = 0$ iff $2^{P-(\mu+1)}$ divides $man\text{-}unrounded$, and (a) follows from Lemma 2.14.

Similarly, it may be shown that $inexact = 0$ iff $2^{P-\mu}$ divides $man\text{-}unrounded$, which implies (b). □

Lemma 3.3

(a) $\textit{man-rounded} = (2^{P-1}\textit{round-carryout}) \mid (\textit{add} \mathbin{\&} \textit{trunc}')$*;*

(b) $\textit{man-rounded}[P-1] = 1$*;*

(c) $\textit{expo}(\textit{man-rounded}) \leq \textit{expo}(\textit{add}) = P - 1 + \textit{round-carryout}$*;*

(d) man-rounded is divisible by 2^{P-M}*.*

Proof: (a) In all cases,

$$\textit{man-rounded} = (2^{P-1}\textit{round-carryout} \mid \textit{add}) \mathbin{\&} \textit{trunc}'$$

and $\textit{trunc}'[P-1] = 1$. Thus, by Lemmas 2.7 and 2.11,

$$\begin{aligned} \textit{man-rounded} &= (2^{P-1}\textit{round-carryout} \mathbin{\&} \textit{trunc}') \mid (\textit{add} \mathbin{\&} \textit{trunc}') \\ &= 2^{P-1}\textit{round-carryout} \mid (\textit{add} \mathbin{\&} \textit{trunc}') \end{aligned}$$

(b) By Lemma 2.10, we may assume $\textit{round-carryout} = 0$ and hence

$$\textit{man-rounded}[P-1] = \textit{add}[P-1].$$

Note that

$$\textit{add} = \begin{cases} \textit{rem}(\textit{man-unrounded} + \textit{rconst} + 1, 2^{P+1}) & \text{if } \textit{op} = \texttt{OP-BACK} \\ \textit{rem}(\textit{man-unrounded} + \textit{rconst}, 2^{P+1}) & \text{otherwise,} \end{cases}$$

and that since

$$\textit{man-unrounded} + \textit{rconst} + 1 \leq (2^P - 1) + (2^P - 1) + 1 < 2^{P+1},$$

we have

$$2^{P-1} \leq \textit{man-unrounded} \leq \textit{add} < 2^{P+1}.$$

But since $\textit{round-carryout} = \textit{add}[P] = 0$, Lemma 2.2 implies $\textit{add} < 2^P$ and hence $\textit{add}[P-1] = 1$.

(c) If $\textit{round-carryout} = 0$, then

$$\textit{man-rounded} = \textit{add} \mathbin{\&} \textit{trunc}' \leq \textit{add} < 2^P,$$

by Lemma 2.9, and $\textit{man-rounded}[P-1] = 1$ implies $\textit{man-rounded} \geq 2^{p-1}$, hence

$$\textit{expo}(\textit{man-rounded}) = \textit{expo}(\textit{add}) = P - 1.$$

On the other hand, if $\textit{round-carryout} = \textit{add}[P] = 1$, then $\textit{expo}(\textit{add}) = P$, while $\textit{man-rounded} < 2^{P+1}$ by Lemma 2.8, hence $\textit{expo}(\textit{man-rounded}) \leq P$.

(d) Since 2^{P-M} divides *trunc*, the result follows from Lemmas 2.9 and 2.8. □

Lemma 3.4 *z is a normal encoding and*

(a) $\textit{sgn}(\hat{z}) = \textit{sgn}(\hat{x}\hat{y})$*;*

(b) $\textit{sig}(\hat{z}) = \textit{rem}(\textit{man-rounded}, 2^P)/2^{P-1}$*;*

(c) $\textit{rem}(\textit{expo}(\hat{z}), 2^{18}) = \textit{rem}(\textit{expo}(\hat{x}\hat{y}) + \textit{round-carryout}, 2^{18})$*.*

Proof: First, observe that

$$z = (\mathit{sign}, \mathit{man\text{-}rounded}[P-1:P-M], \mathit{exp\text{-}rounded}).$$

Let $\rho = \mathit{rem}(\mathit{man\text{-}rounded}, 2^P)$. By Lemma 2.3,

$$\rho[P-1] = \mathit{man\text{-}rounded}[P-1] = 1,$$

hence $\mathit{expo}(\rho) = P-1$. Since *man-rounded* is divisible by 2^{P-M}, so is ρ. Thus, by Lemmas 2.4 and 2.14,

$$\mathit{get\text{-}man}(z) = \mathit{man\text{-}rounded}[P-1:P-M] = \rho[P-1:P-M] = \rho/2^{P-M}.$$

It follows that

$$\mathit{expo}(\mathit{get\text{-}man}(z)) = \mathit{expo}(\rho) - (P-M) = (P-1) - (P-M) = M-1.$$

Since

$$\mathit{get\text{-}expo}(z) = \mathit{exp\text{-}rounded} = \mathit{rem}(\mathit{exp\text{-}unrounded} + \mathit{round\text{-}carryout} + \mathit{overflow}, 2^{18}),$$

$0 < \mathit{get\text{-}expo}(z) < 2^{18}$, and hence z is a normal encoding. The proof is completed by applying Lemma 2.15:

(a) $\mathit{sgn}(\hat{z}) = (-1)^{\mathit{sign}}$, hence $\mathit{sgn}(\hat{z}) = 1 \Leftrightarrow \mathit{sign} = 0 \Leftrightarrow \mathit{get\text{-}sign}(x) = \mathit{get\text{-}sign}(y) \Leftrightarrow \mathit{sgn}(\hat{x}) = \mathit{sgn}(\hat{y}) \Leftrightarrow \mathit{sgn}(\hat{x}\hat{y}) = 1$.

(b) $\mathit{sig}(\hat{z}) = \mathit{get\text{-}man}(z)/2^{M-1} = (\rho/2^{P-M})/2^{M-1} = \rho/2^{P-1}$.

(c) $\mathit{expo}(\hat{z}) = \mathit{get\text{-}expo}(z) - (2^{17}-1)$, where

$$\begin{aligned}
&\mathit{get\text{-}expo}(z)\\
&\quad = \mathit{rem}(\mathit{exp\text{-}unrounded} + \mathit{round\text{-}carryout} + \mathit{overflow}, 2^{18})\\
&\quad = \mathit{rem}(\mathit{get\text{-}expo}(x) + \mathit{get\text{-}expo}(y) + 2^{17} + 1 + \mathit{round\text{-}carryout} + \mathit{overflow}, 2^{18})\\
&\quad = \mathit{rem}(\mathit{expo}(\hat{x}) + \mathit{expo}(\hat{y}) + 2^{18} - 2 + 2^{17} + 1 + \mathit{round\text{-}carryout} + \mathit{overflow}, 2^{18})\\
&\quad = \mathit{rem}(\mathit{expo}(\hat{x}) + \mathit{expo}(\hat{y}) + \mathit{overflow} + 2^{17} - 1 + \mathit{round\text{-}carryout}, 2^{18})\\
&\quad = \mathit{rem}(\mathit{expo}(\hat{x}\hat{y}) + 2^{17} - 1 + \mathit{round\text{-}carryout}, 2^{18}).\Box
\end{aligned}$$

3.3 The Operations OP-MUL, OP-DIV, and OP-SQRT

This is our statement of IEEE compliance for multiplication:

Theorem 1 *Assume that op* $\in \{\texttt{OP-MUL}, \texttt{OP-DIV}, \texttt{OP-SQRT}\}$*, rc is a rounding control specifier, pc is a precision control specifier, and x and y are normal encodings. If* $\mathit{rnd}(\hat{x}\hat{y}, \mathit{rc}, \mathit{pc})$ *is representable, then* $\hat{z} = \mathit{rnd}(\hat{x}\hat{y}, \mathit{rc}, \mathit{pc})$.

Proof: Let

$$\mathit{rc}' = \begin{cases} \texttt{RC-NEG} & \text{if } \mathit{rc} = \texttt{RC-POS}\\ \texttt{RC-POS} & \text{if } \mathit{rc} = \texttt{RC-NEG}\\ \mathit{rc} & \text{otherwise.} \end{cases}$$

Then $\mathit{rnd}(-\hat{x}\hat{y}, \mathit{rc}, \mathit{pc}) = -\mathit{rnd}(\hat{x}\hat{y}, \mathit{rc}', \mathit{pc})$. Also, by inspection of the code that defines *FPU-MUL*, it is easy to see that replacing either *get-sign(x)* or *get-sign(y)* by its complement and rc by rc' has the effect of negating $\hat{z}$. It follows that it suffices to prove the theorem under the assumptions $\hat{x} > 0$ and $\hat{y} > 0$, which imply that $\mathit{sign} = 0$.

Note that (under these assumptions)

$$rconst = \begin{cases} 2^{P-\mu-1} & \text{if } rc = \texttt{RC-NEAR} \\ 2^{P-\mu}-1 & \text{if } rc = \texttt{RC-POS} \\ 0 & \text{otherwise.} \end{cases}$$

In all cases, $rconst < 2^P$. Since *man-unrounded* $< 2^P$ as well,

$$add = rem(\mathit{man\text{-}unrounded} + rconst, 2^{P+1}) = \mathit{man\text{-}unrounded} + rconst.$$

If $rc = \texttt{RC-NEAR}$ and $sticky = add[P-\mu-1] = 0$, then by Lemma 2.12,

$$trunc' = (2^{2M} - 2^{P-\mu}) \,\&\, (2^{2M} - 1 - 2^{P-\mu}) = (2^{2M} - 2^{P-\mu+1}),$$

and otherwise

$$trunc' = (2^{2M} - 2^{P-\mu}).$$

We shall show that

$$rem(\mathit{man\text{-}rounded}, 2^P) = rnd(\mathit{man\text{-}unrounded}, rc, pc)2^{-\mathit{round\text{-}carryout}},$$

by considering the following cases:

Case 1: round-carryout $= 0$

Since *man-rounded* $< 2^P$ by Lemma 3.3, we must show

$$\mathit{man\text{-}rounded} = rnd(\mathit{man\text{-}unrounded}, rc, pc).$$

Subcase 1.1: $rc = \texttt{RC-NEAR}$

First suppose $sticky = add[P-\mu-1] = 0$. Then Lemma 2.3 implies

$$\mathit{man\text{-}unrounded}[P-\mu-1] = 1,$$

and by Lemmas 3.2 and 2.14, *man-unrounded* is $(\mu+1)$-exact but not μ-exact. Thus, by Lemmas 3.3, 2.24, and 2.31,

$$\begin{aligned} \mathit{man\text{-}rounded} &= (\mathit{man\text{-}unrounded} + 2^{P-\mu-1}) \,\&\, (2^{2M} - 2^{P-\mu+1}) \\ &= trunc(\mathit{man\text{-}unrounded} + 2^{P-\mu-1}, \mu - 1) \\ &= near(\mathit{man\text{-}unrounded}, \mu) \\ &= rnd(\mathit{man\text{-}unrounded}, rc, pc). \end{aligned}$$

In the remaining case, *man-unrounded* is either μ-exact or not $(\mu+1)$-exact, and the same three lemmas yield

$$\begin{aligned} \mathit{man\text{-}rounded} &= (\mathit{man\text{-}unrounded} + 2^{P-\mu-1}) \,\&\, (2^{2M} - 2^{P-\mu}) \\ &= trunc(\mathit{man\text{-}unrounded} + 2^{P-\mu-1}, \mu) \\ &= near(\mathit{man\text{-}unrounded}, \mu) \\ &= rnd(\mathit{man\text{-}unrounded}, rc, pc). \end{aligned}$$

Subcase 1.2: $rc = \texttt{RC-POS}$

By Lemmas 2.24 and 2.25,

$$\begin{aligned} \textit{man-rounded} &= (\textit{man-unrounded} + 2^{P-\mu} - 1) \mathbin{\&} (2^{2M} - 2^{P-\mu}) \\ &= \textit{trunc}(\textit{man-unrounded} + 2^{P-\mu} - 1, \mu) \\ &= \textit{away}(\textit{man-unrounded}, \mu) \\ &= \textit{rnd}(\textit{man-unrounded}, rc, pc). \end{aligned}$$

Subcase 1.3: $rc = \texttt{RC-CHOP}$ or $rc = \texttt{RC-NEG}$
By Lemma 2.24,

$$\begin{aligned} \textit{man-rounded} &= \textit{man-unrounded} \mathbin{\&} (2^{2M} - 2^{P-\mu}) \\ &= \textit{trunc}(\textit{man-unrounded}, \mu) \\ &= \textit{rnd}(\textit{man-unrounded}, rc, pc). \end{aligned}$$

Case 2: round-carryout $= 1$
In this case,

$$2^P \leq add = \textit{man-unrounded} + rconst < 2^P + rconst,$$

which implies

$$0 \leq rem(add, 2^P) < rconst < 2^{P-\mu}.$$

By Lemmas 3.3, 2.9, and 2.8,

$$\begin{aligned} rem(\textit{man-rounded}, 2^P) &= rem(2^{P-1} \mid (add \mathbin{\&} trunc'), 2^P) \\ &= 2^{P-1} \mid (rem(add, 2^P) \mathbin{\&} trunc') \\ &= 2^{P-1} \mid (rem(add, 2^P) \mathbin{\&} rem(trunc', 2^{P-\mu})) \\ &= 2^{P-1} \mid (rem(add, 2^P) \mathbin{\&} 0) \\ &= 2^{P-1}. \end{aligned}$$

Thus, it suffices to show that $rnd(\textit{man-unrounded}, rc, pc) = 2^P$.

Subcase 2.1: $rc = \texttt{RC-NEAR}$
Since

$$\textit{man-unrounded} + 2^{P-1-\mu} = \textit{man-unrounded} + rconst \geq 2^P,$$

$near(\textit{man-unrounded}, \mu) = 2^P$ by Lemma 2.30.

Subcase 2.2: $rc = \texttt{RC-POS}$
Let $a = 2^P - 2^{P-\mu}$. Then

$$\textit{man-unrounded} \geq 2^P - rconst = 2^P - 2^{P-\mu} + 1 > a,$$

and since a is μ-exact,

$$away(\textit{man-unrounded}, \mu) \geq a + 2^{expo(a)+1-\mu} = a + 2^{P-\mu} = 2^P,$$

which implies $away(\textit{man-unrounded}, \mu) = 2^P$.

Subcase 2.3: $rc = \texttt{RC-CHOP}$ or $rc = \texttt{RC-NEG}$
This case is precluded by our earlier observation that $0 < rconst$.

The proof is completed by applying Lemmas 3.4 and 3.1, which yield

$$sgn(\hat{z}) = sgn(\hat{x}\hat{y}) = 1,$$

$$\begin{aligned} sig(\hat{z}) &= rnd(man\text{-}unrounded, rc, pc)2^{-round\text{-}carryout-P+1} \\ &= rnd(sig(\hat{x}\hat{y}), rc, pc)2^{-round\text{-}carryout}, \end{aligned}$$

and for some $k \in \mathbb{Z}$,

$$expo(\hat{z}) = expo(\hat{x}\hat{y}) + round\text{-}carryout + 2^{18}k.$$

Thus,

$$\hat{z} = rnd(sig(\hat{x}\hat{y}), rc, pc)2^{expo(\hat{x}\hat{y})+2^{18}k} = rnd(\hat{x}\hat{y}, rc, pc)2^{2^{18}k}.$$

But since $rnd(\hat{x}\hat{y}, rc, pc)$ is representable, i.e., $1 - 2^{-17} \leq expo(rnd(\hat{x}\hat{y}, rc, pc)) \leq 2^{17}$, and the same is true of $\hat{z}$,

$$|2^{18}k| = |expo(\hat{z}) - expo(rnd(\hat{x}\hat{y}, rc, pc))| < 2^{18},$$

and hence $k = 0$. □

In the `OP-DIV` and `OP-SQRT` cases, an additional value is returned:

Lemma 3.5 *Let $op \in \{$`OP-DIV`, `OP-SQRT`$\}$, $pc =$ `PC-*`, and $rc =$ `RC-NEAR`. Assume that x and y are normal encodings, $3/2 < sig(\hat{x})sig(\hat{y}) < 3$, and $|1 - \hat{x}\hat{y}| < 1/8$. Then*

(a) r is a normal encoding;
(b) $\hat{r} < 1 \Leftrightarrow \hat{z} \geq 1$;
(c) if $op =$ `OP-DIV`, then $2 - \hat{x}\hat{y} - 2^{1-M} \leq \hat{r} < 2 - \hat{x}\hat{y}$;
(d) if $op =$ `OP-SQRT`, then $(3 - \hat{x}\hat{y})/2 - 2^{1-M} \leq \hat{r} < (3 - \hat{x}\hat{y})/2$.

Proof: First note that the hypothesis implies that $expo(\hat{x}\hat{y})$ is either 0 or -1, and it follows from Lemma 3.4 that

$$expo(\hat{z}) = expo(\hat{x}\hat{y}) + round\text{-}carryout.$$

We consider the following cases:

Case 1: $overflow = 1$

In this case, $expo(man\text{-}unrounded) = 2M - 1$, but by Lemma 3.1,

$$man\text{-}unrounded = sig(\hat{x}\hat{y})2^{2M-1} = sig(\hat{x})sig(\hat{y})2^{2M-2} < 3 \cdot 2^{2M-2}$$

and hence

$$add = man\text{-}unrounded + 2^{M-1} < 3 \cdot 2^{2M-2} + 2^{M-1} < 2^{2M}$$

and $round\text{-}carryout = 0$. We have $expo(\hat{z}) = expo(\hat{x}\hat{y}) = 0$, for otherwise

$$\hat{x}\hat{y} = sig(\hat{x}\hat{y})/2 = sig(\hat{x})sig(\hat{y})/4 < 3/4,$$

contradicting $|1 - \hat{x}\hat{y}| < 1/8$. Thus,

$$\hat{x}\hat{y} = sig(\hat{x}\hat{y}) = sig(man\text{-}unrounded) = man\text{-}unrounded/2^{2M-1}.$$

Also note that

$$\begin{aligned}
\mathit{comp1}(\mathit{man\text{-}unrounded}, 2M) &= 2^{2M} - \mathit{man\text{-}unrounded} - 1 \\
&\leq 2^{2M} - 2^{2M-1} - 1 < 2^{2M-1}.
\end{aligned}$$

Subcase 1.1: $\mathit{op} = \texttt{OP-DIV}$

$$\begin{aligned}
\mathit{get\text{-}man}(r) &= \mathit{comp1}(\mathit{man\text{-}unrounded}, 2M)[2M-2 : M-1] \\
&= \lfloor (2^{2M} - \mathit{man\text{-}unrounded} - 1)/2^{M-1} \rfloor \\
&= 2^{M+1} + \lfloor -(\mathit{man\text{-}unrounded} + 1)/2^{M-1} \rfloor \\
&= 2^{M+1} - \lfloor \mathit{man\text{-}unrounded}/2^{M-1} \rfloor - 1.
\end{aligned}$$

But

$$\lfloor \mathit{man\text{-}unrounded}/2^{M-1} \rfloor \leq \mathit{man\text{-}unrounded}/2^{M-1} = 2^M \hat{x}\hat{y}$$

and

$$\lfloor \mathit{man\text{-}unrounded}/2^{M-1} \rfloor > \mathit{man\text{-}unrounded}/2^{M-1} - 1 = 2^M \hat{x}\hat{y} - 1,$$

hence

$$2^{M-1} \leq 2^{M+1} - 2^M \hat{x}\hat{y} - 1 \leq \mathit{get\text{-}man}(r) < 2^{M+1} - 2^M \hat{x}\hat{y} \leq 2^M$$

and r is normal. Since $\mathit{expo}(\hat{r}) = 2^{17} - 2 - (2^{17} - 1) = -1$, $\hat{r} < 1 \leq \hat{z}$ and

$$2 - \hat{x}\hat{y} - 2^{-M} \leq \hat{r} = 2^{-M}\mathit{get\text{-}man}(r) < 2 - \hat{x}\hat{y}.$$

Subcase 1.2: $\mathit{op} = \texttt{OP-SQRT}$

By Lemmas 2.2 and 2.11,

$$\begin{aligned}
\mathit{comp1}(\mathit{man\text{-}unrounded}, 2M) \mid 2^{2M-1} &= \mathit{comp1}(\mathit{man\text{-}unrounded}, 2M) + 2^{2M-1} \\
&= 2^{2M} + 2^{2M-1} - \mathit{man\text{-}unrounded} - 1 \\
&< 2^{2M},
\end{aligned}$$

hence

$$\begin{aligned}
\mathit{get\text{-}man}(r) &= (\mathit{comp1}(\mathit{man\text{-}unrounded}, 2M) \mid 2^{2M-1})[2M-1 : M] \\
&= (2^{2M} + 2^{2M-1} - \mathit{man\text{-}unrounded} - 1)[2M-1 : M] \\
&= \lfloor (2^{2M} + 2^{2M-1} - \mathit{man\text{-}unrounded} - 1)/2^M \rfloor \\
&= 2^M + 2^{M-1} + \lfloor -(\mathit{man\text{-}unrounded} + 1)/2^M \rfloor \\
&= 3 \cdot 2^{M-1} - \lfloor \mathit{man\text{-}unrounded}/2^M \rfloor - 1.
\end{aligned}$$

But

$$\lfloor \mathit{man\text{-}unrounded}/2^M \rfloor \leq \mathit{man\text{-}unrounded}/2^M = 2^{M-1}\hat{x}\hat{y}$$

and

$$\lfloor \mathit{man\text{-}unrounded}/2^M \rfloor > \mathit{man\text{-}unrounded}/2^M - 1 = 2^{M-1}\hat{x}\hat{y} - 1,$$

implying

$$2^{M-1} \leq 2^{M-1}(3 - \hat{x}\hat{y}) - 1 \leq \mathit{get\text{-}man}(r) < 2^{M-1}(3 - \hat{x}\hat{y}) \leq 2^M,$$

hence r is normal. Again, $\mathit{expo}(\hat{r}) = -1$ and $\hat{r} < 1 \leq \hat{z}$. Thus

$$(3 - \hat{x}\hat{y})/2 - 2^{-M} \leq \hat{r} = \mathit{get\text{-}man}(r)/2^M < (3 - \hat{x}\hat{y})/2.$$

Case 2: $overflow = 0$

In this case, $\textit{expo}(\textit{man-unrounded}) = 2M - 2$, and $\textit{expo}(\hat{x}\hat{y}) = -1$, for otherwise

$$\hat{x}\hat{y} = \textit{sig}(\hat{x}\hat{y}) = \textit{sig}(\hat{x})\textit{sig}(\hat{y}) > 3/2,$$

contradicting $|1 - \hat{x}\hat{y}| < 1/8$. Thus

$$\hat{x}\hat{y} = \textit{sig}(\hat{x}\hat{y})/2 = \textit{sig}(\textit{man-unrounded})/2 = \textit{man-unrounded}/2^{2M-1}.$$

Subcase 2.1: $\textit{round-carryout} = 0$

Since $\textit{expo}(\hat{z}) = \textit{expo}(\hat{x}\hat{y}) = -1$, $\hat{z} < 1$.

Subcase 2.1.1: $\textit{op} = \texttt{OP-DIV}$

$$\begin{aligned}
\textit{get-man}(r) &= (2^{2M} - \textit{man-unrounded} - 1)[2M-1:M] \\
&= \lfloor (2^{2M} - \textit{man-unrounded} - 1)/2^M \rfloor \\
&= 2^M + \lfloor -(\textit{man-unrounded} + 1)/2^M \rfloor \\
&= 2^M - \lfloor \textit{man-unrounded}/2^M \rfloor - 1.
\end{aligned}$$

In this case,

$$2^{M-1}\hat{x}\hat{y} - 1 < \lfloor \textit{man-unrounded}/2^M \rfloor \leq 2^{M-1}\hat{x}\hat{y}$$

and

$$2^{M-1} - 1 < 2^M - 2^{M-1}\hat{x}\hat{y} - 1 \leq \textit{get-man}(r) < 2^M - 2^{M-1}\hat{x}\hat{y} < 2^M.$$

Since $\textit{expo}(\hat{r}) = (2^{17} - 1) - (2^{17} - 1) = 0$, $\hat{r} \geq 1 > \hat{z}$, and

$$2 - \hat{x}\hat{y} - 2^{1-M} \leq \hat{r} = \textit{get-man}(r)/2^{M-1} < 2 - \hat{x}\hat{y}.$$

Subcase 2.1.2: $\textit{op} = \texttt{OP-SQRT}$

Note that

$$\textit{comp1}(\textit{man-unrounded}, 2M) = 2^{2M} - \textit{man-unrounded} - 1 \geq 2^{2M} - 2^{2M-1} = 2^{2M-1},$$

while $\textit{comp1}(\textit{man-unrounded}, 2M) < 2^{2M}$, hence

$$\begin{aligned}
&\textit{rem}(\textit{comp1}(\textit{man-unrounded}, 2^{2M}), 2^{2M-1}) \\
&\quad = \textit{comp1}(\textit{man-unrounded}, 2M) - 2^{2M-1} \\
&\quad = 2^{2M-1} - \textit{man-unrounded} - 1.
\end{aligned}$$

Therefore, applying Lemma 2.11, we have

$$\begin{aligned}
\textit{get-man}(r) &= \textit{shr}(\textit{comp1}(\textit{man-unrounded}, 2M)[2M-2:0], 1, 2M)[2M-1:M] \\
&= \textit{shr}(\textit{rem}(\textit{comp1}(\textit{man-unrounded}, 2M), 2^{2M-1}), 1, 2M)[2M-1:M] \\
&= \textit{shr}(2^{2M-1} - \textit{man-unrounded} - 1, 1, 2M)[2M-1:M] \\
&= (2^{2M-1} + \lfloor (2^{2M-1} - \textit{man-unrounded} - 1)/2 \rfloor)[2M-1:M] \\
&= \lfloor (2^{2M-1} + \lfloor (2^{2M-1} - \textit{man-unrounded} - 1)/2 \rfloor)/2^M \rfloor \\
&= 2^{M-1} + \lfloor \lfloor (2^{2M-1} - \textit{man-unrounded} - 1)/2 \rfloor)/2^M \rfloor \\
&= 2^{M-1} + \lfloor (2^{2M-1} - \textit{man-unrounded} - 1)/2^{M+1} \rfloor \\
&= 2^{M-1} + 2^{M-2} + \lfloor -(\textit{man-unrounded} + 1)/2^{M+1} \rfloor) \\
&= 3 \cdot 2^{M-2} - \lfloor \textit{man-unrounded}/2^{M+1} \rfloor - 1.
\end{aligned}$$

But

$$2^{M-2}\hat{x}\hat{y} - 1 < \lfloor \mathit{man\text{-}unrounded}/2^{M+1} \rfloor \leq 2^{M-2}\hat{x}\hat{y},$$

hence

$$2^{M-1} - 1 < 2^{M-2}(3 - \hat{x}\hat{y}) - 1 \leq \mathit{get\text{-}man}(r) < 2^{M-2}(3 - \hat{x}\hat{y}) < 2^{M}.$$

Again, $\mathit{expo}(\hat{r}) = 0$, $\hat{r} \geq 1 > \hat{z}$, and

$$(3 - \hat{x}\hat{y})/2 - 2^{1-M} \leq \hat{r} = \mathit{get\text{-}man}(r)/2^{M-1} < (3 - \hat{x}\hat{y})/2.$$

Subcase 2.2: round-carryout $= 1$

In this case, $\mathit{get\text{-}man}(r) = 2^{M} - 1$ and $\hat{r} = 1 - 2^{-M} < 1$, while $\mathit{expo}(\hat{z}) = \mathit{expo}(\hat{x}\hat{y}) + 1 = 0$, so $\hat{z} \geq 1$. Since $\mathit{add} = \mathit{man\text{-}unrounded} + 2^{M-2} \geq 2^{2M-1}$, we have

$$2^{2M-1} - 2^{M-2} \leq \mathit{man\text{-}unrounded} < 2^{2M-1}$$

and hence

$$1 - 2^{-1-M} \leq \hat{x}\hat{y} < 1,$$

which implies

$$2 - \hat{x}\hat{y} - (2^{-M} + 2^{-1-M}) \leq \hat{r} < 2 - \hat{x}\hat{y} - 2^{-M}$$

and

$$(3 - \hat{x}\hat{y})/2 - (2^{-M} + 2^{-2-M}) \leq \hat{r} < (3 - \hat{x}\hat{y})/2 - 2^{-M}.\Box$$

The following corollary of Lemma 3.5 allows the outputs of *FPU-MUL* to be used as inputs on the next iteration of *FPU-DIV-SQRT*:

Lemma 3.6 *Let op* $\in \{$`OP-DIV`, `OP-SQRT`$\}$, *pc* $=$ `PC-*`, *and rc* $=$ `RC-NEAR`. *Assume that x and y are normal encodings,* $3/2 < \mathit{sig}(\hat{x})\mathit{sig}(\hat{y}) < 3$, *and* $|1 - \hat{x}\hat{y}| < 1/8$. *Then*

(a) if op $=$ `OP-DIV`, *then* $3/2 < \mathit{sig}(\hat{z})\mathit{sig}(\hat{r}) < 3$*;*

(b) if op $=$ `OP-SQRT`, *then* $3/2 < \mathit{sig}(\hat{z})\mathit{sig}(\mathit{near}(\hat{r}^2, M)) < 3$*;*

Proof: Note first that by Theorem 1, $|1 - \hat{z}| \leq 1/8$. Now suppose that $\hat{z} < 1$. Then $7/8 \leq \hat{z} < 1$. If *op* $=$ `OP-DIV`, then $1 \leq \hat{r} < 2 - \hat{x}\hat{y} \leq 9/8$, hence $\mathit{sig}(\hat{z})\mathit{sig}(\hat{r}) = 2\hat{z}\hat{r}$ and $3/2 < 7/4 \leq 2\hat{z}\hat{r} < 9/4 < 3$. For the case *op* $=$ `OP-SQRT`, let $w = \mathit{near}(\hat{r}^2, M)$. Since $1 \leq \hat{r} < (3 - \hat{x}\hat{y})/2 < 17/16$, $1 \leq \hat{r}^2 < 289/256 < 3/2$, which implies $1 \leq w < 3/2$. Thus, $\mathit{sig}(\hat{z})\mathit{sig}(w) = 2\hat{z}w$ and $3/2 < 7/4 < 2\hat{z}w < 3$.

On the other hand, if $\hat{z} \geq 1$, then $1 \leq \hat{z} < 9/8$. If *op* $=$ `OP-DIV`, then $1 > \hat{r} \geq 2 - \hat{x}\hat{y} - 2^{1-M} \geq 7/8 - 2^{1-M} > 3/4$, and again $\mathit{sig}(\hat{z})\mathit{sig}(\hat{r}) = 2\hat{z}\hat{r}$, where $3/2 < 2\hat{z}\hat{r} < 9/4 < 3$. If *op* $=$ `OP-SQRT`, then $1 > \hat{r} \geq (3 - \hat{x}\hat{y})/2 - 2^{1-M} \geq 15/16 - 2^{1-M} > 7/8$ and $1 > \hat{r}^2 > 49/64$, which implies $1 >\geq w \geq 49/64 > 3/4$. Thus, $\mathit{sig}(\hat{z})\mathit{sig}(w) = 2\hat{z}w$ and $3/2 < 2\hat{z}w \leq 9/4 < 3.\Box$

3.4 The Operation `OP-LAST`

In the `OP-LAST` case, the product is rounded to $\mathit{mbits}(\mathit{lastpc}) + 1$ bits, essentially by *near* rounding:

Lemma 3.7 *If $op = \texttt{OP-LAST}$, $pc = \texttt{PC-*}$, $rc = \texttt{RC-NEAR}$, $mbits(lastpc) = \lambda$, x and y are normal encodings, and*

$$2^{-2^{17}}(2 - 2^{-\lambda-1}) \leq |\hat{x}\hat{y}| < 2^{2^{17}}(2 - 2^{-\lambda-1}),$$

then

(a) $\hat{z}$ is $(\lambda+1)$-exact; (b) $expo(\hat{x}\hat{y}) \leq expo(\hat{z})$; (c) $|\hat{z} - \hat{x}\hat{y}| \leq 2^{expo(\hat{x}\hat{y})-\lambda-1}$.

Proof: Note that

$$add = man\text{-}unrounded + 2^{P-\lambda-2}$$

and by Lemma 2.12,

$$trunc = 2^{2M} - 2^{P-\lambda-1} = trunc'.$$

Let $\rho = rem(man\text{-}rounded, 2^P)$. We shall show that

$$|\rho 2^{round\text{-}carryout} - man\text{-}unrounded| \leq 2^{P-\lambda-2}$$

and that

$$1 - 2^{17} \leq expo(\hat{x}\hat{y}) + round\text{-}carryout \leq 2^{17},$$

by considering the following two cases:

Case 1: $round\text{-}carryout = 0$

By Lemma 3.3, $expo(add) = expo(man\text{-}rounded) = P - 1$, hence

$$\rho = man\text{-}rounded = add \,\&\, (2^{2M} - 2^{P-\lambda-1}) = trunc(add, \lambda+1)$$

by Lemma 2.24. Thus, by Lemma 2.20,

$$\rho \leq add = man\text{-}unrounded + 2^{P-\lambda-2}$$

and

$$\rho > add - 2^{(P-1)-(\lambda+1)+1} = man\text{-}unrounded - 2^{P-\lambda-2}.$$

If $expo(\hat{x}\hat{y}) = 2^{-17}$, then $2^{-2^{17}}(2 - 2^{-\lambda-1}) \leq |\hat{x}\hat{y}| < 2^{-2^{17}+1}$, hence

$$man\text{-}unrounded = 2^{P-1}sig(\hat{x}\hat{y}) \geq 2^{P-1}(2 - 2^{-\lambda-1}) = 2^P - 2^{P-\lambda-2},$$

contradicting $add < 2^P$. Thus, $1 - 2^{17} \leq expo(\hat{x}\hat{y}) \leq 2^{17}$.

Case 2: $round\text{-}carryout = 1$

In this case,

$$2^P \leq add = man\text{-}unrounded + 2^{P-\lambda-2} < 2^P + 2^{P-\lambda-2},$$

which implies

$$|2^P - man\text{-}unrounded| < 2^{P-\lambda-2}$$

as well as

$$rem(add, 2^P) < 2^{P-\lambda-2}.$$

Thus, by Lemmas 3.3, 2.9, 2.8, and 2.7,

$$\begin{aligned}
\rho &= rem(2^{P-1} \mid (add \,\&\, trunc'), 2^P) = 2^{P-1} \mid (rem(add, 2^P) \,\&\, trunc') \\
&= 2^{P-1} \mid (rem(add, 2^P) \,\&\, rem(trunc', 2^{P-\lambda-2})) = 2^{P-1} \mid (rem(add, 2^P) \,\&\, 0) \\
&= 2^{P-1},
\end{aligned}$$

and therefore

$$|2\rho - \mathit{man\text{-}unrounded}| = |2^P - \mathit{man\text{-}unrounded}| < 2^{P-\lambda-2}.$$

If $\mathit{expo}(\hat{x}\hat{y}) = 2^{17}$, then $2^{2^{17}} \leq |\hat{x}\hat{y}| < 2^{2^{17}}(2 - 2^{-\lambda-1})$, hence

$$\mathit{man\text{-}unrounded} = 2^{P-1}\mathit{sig}(\hat{x}\hat{y}) < 2^{P-1}(2 - 2^{-\lambda-1}) = 2^P - 2^{P-\lambda-2},$$

contradicting $\mathit{add} \geq 2^P$. Thus, $1 - 2^{17} \leq \mathit{expo}(\hat{x}\hat{y}) + 1 \leq 2^{17}$.

Note that in both cases, ρ is $(\lambda+1)$-exact, hence so is $\hat{z}$, since $\mathit{sig}(\hat{z}) = \rho 2^{1-P}$. Since

$$1 - 2^{17} \leq \mathit{expo}(\hat{x}\hat{y}) + \mathit{round\text{-}carryout} \leq 2^{17},$$

and $\mathit{expo}(\hat{z})$ must lie in the same interval,

$$\mathit{expo}(\hat{z}) = \mathit{expo}(\hat{x}\hat{y}) + \mathit{round\text{-}carryout}.$$

Thus,

$$\begin{aligned} |\hat{z} - \hat{x}\hat{y}| &= |\rho 2^{1-P} 2^{\mathit{expo}(\hat{x}\hat{y}) + \mathit{round\text{-}carryout}} - \mathit{sig}(\hat{x}\hat{y}) 2^{\mathit{expo}(\hat{x}\hat{y})}| \\ &= 2^{\mathit{expo}(\hat{x}\hat{y})+1-P} |\rho 2^{\mathit{round\text{-}carryout}} - \mathit{man\text{-}unrounded}| \\ &\leq 2^{\mathit{expo}(\hat{x}\hat{y})+1-P} 2^{P-\lambda-2} \\ &= 2^{\mathit{expo}(\hat{x}\hat{y})-\lambda-1}. \square \end{aligned}$$

3.5 The Operation `OP-BACK`

In the `OP-BACK` case, the product is compared, by way of subtraction, to the input d. The results of the comparison are given by the outputs z and *inexact*:

Lemma 3.8 *If* $\mathit{op} = $ `OP-BACK`, $\mathit{pc} = $ `PC-*`, $\mathit{rc} = $ `RC-CHOP`, *x and y are normal encodings, and* $|\hat{x}\hat{y} - \hat{d}| < 2^{\mathit{expo}(\hat{d})-3}$, *then*

(a) $|\hat{x}\hat{y}| < |\hat{d}| \Leftrightarrow \mathit{get\text{-}man}(z)[M-2] = 1$;

(b) $\hat{x}\hat{y} = \hat{d} \Leftrightarrow \mathit{get\text{-}man}(z)[M-2:0] = \mathit{inexact} = 0$.

Proof: (a) Since

$$\begin{aligned} \mathit{rconst\text{-}with\text{-}overflow} &= \mathit{comp1}(2^M \mathit{get\text{-}man}(d), 2M) \\ &= 2^{2M} - 2^M \mathit{get\text{-}man}(d) - 1 \end{aligned}$$

and

$$\begin{aligned} \mathit{rconst\text{-}no\text{-}overflow} &= \mathit{shr}(\mathit{rconst\text{-}with\text{-}overflow}, 0, 2M) \\ &= \lfloor (2^{2M} - 2^M \mathit{get\text{-}man}(d) - 1)/2 \rfloor \\ &= 2^{2M-1} - 2^{M-1} \mathit{get\text{-}man}(d) - 1, \end{aligned}$$

we have

$$\mathit{rconst} = 2^P - 2^{P-M} \mathit{get\text{-}man}(d) - 1,$$

and thus

$$\begin{aligned} \mathit{add} &= \mathit{rem}(2^P + \mathit{man\text{-}unrounded} - 2^{P-M}\mathit{get\text{-}man}(d), 2^{P+1}) \\ &= \mathit{rem}(2^P + 2^{P-1}\mathit{sig}(\hat{x}\hat{y}) - 2^{P-1}\mathit{sig}(\hat{d}), 2^{P+1}) \\ &= \mathit{rem}(2^{P-1}(2 + \mathit{sig}(\hat{x}\hat{y}) - \mathit{sig}(\hat{d})), 2^{P+1}) \\ &= 2^{P-1}(2 + \mathit{sig}(\hat{x}\hat{y}) - \mathit{sig}(\hat{d})). \end{aligned}$$

Note also that $\mathit{trunc}' = \mathit{trunc} = 2^{2M} - 2^{P-M}$.

By Lemmas 2.4, 2.5, 2.11, and 3.3,

$$\begin{aligned} \mathit{get\text{-}man}(z)[M-2:0] &= (\mathit{man\text{-}rounded}[P-1:P-M])[M-2:0] \\ &= \mathit{man\text{-}rounded}[P-2:P-M] \\ &= (\mathit{add}\ \&\ \mathit{trunc}')[P-2:P-M] \\ &= (2^{P-M}\mathit{add}[2M-1:P-M])[P-2:P-M] \\ &= \mathit{add}[2M-1:P-M][M-2:0] \\ &= \mathit{add}[P-2:P-M] \\ &= \rho[P-2:P-M], \end{aligned}$$

where $\rho = \mathit{rem}(\mathit{add}, 2^{P-1})$. In particular, by Lemma 2.5,

$$\begin{aligned} \mathit{get\text{-}man}(z)[M-2] &= \mathit{get\text{-}man}(z)[M-2:0][M-2] \\ &= \rho[P-2:P-M][M-2] = \rho[P-2]. \end{aligned}$$

We must show

$$\rho[P-2] = 1 \Leftrightarrow |\hat{x}\hat{y}| < |\hat{d}|.$$

Since

$$|\hat{x}\hat{y} - \hat{d}| = |2^{\mathit{expo}(\hat{x}\hat{y}) - \mathit{expo}(\hat{d})}\mathit{sig}(\hat{x}\hat{y}) - \mathit{sig}(\hat{d})|2^{\mathit{expo}(\hat{d})} < 2^{\mathit{expo}(\hat{d})-3},$$

we have

$$|2^{\mathit{expo}(\hat{x}\hat{y}) - \mathit{expo}(\hat{d})}\mathit{sig}(\hat{x}\hat{y}) - \mathit{sig}(\hat{d})| < 2^{-3},$$

which implies $|\mathit{expo}(\hat{x}\hat{y}) - \mathit{expo}(\hat{d})| \leq 1$. Thus, we have three cases to consider:

Case 1: $\mathit{expo}(\hat{x}\hat{y}) = \mathit{expo}(\hat{d})$

In this case, $|\mathit{sig}(\hat{x}\hat{y}) - \mathit{sig}(\hat{d})| < 2^{-3}$.

Suppose first that $|\hat{x}\hat{y}| < |\hat{d}|$. Then $\mathit{sig}(\hat{x}\hat{y}) < \mathit{sig}(\hat{d})$ and

$$2^P > \mathit{add} = 2^{P-1}(2 + \mathit{sig}(\hat{x}\hat{y}) - \mathit{sig}(\hat{d})) > 2^{P-1}(2 - 2^{-3}) > 2^{P-1} + 2^{P-2}.$$

Thus,

$$2^{P-2} < \rho < 2^{P-1},$$

and $\rho[P-2] = 1$ by Lemma 2.2.

On the other hand, if $|\hat{x}\hat{y}| \geq |\hat{d}|$, then $\mathit{sig}(\hat{x}\hat{y}) \geq \mathit{sig}(\hat{d})$ and

$$2^P \leq \mathit{add} < 2^{P-1}(2 + 2^{-3}) < 2^P + 2^{P-2},$$

hence $\rho < 2^{P-2}$ and $\rho[P-2] = 0$.

Case 2: $expo(\hat{x}\hat{y}) = expo(\hat{d}) + 1$

Here, $|\hat{x}\hat{y}| > |\hat{d}|$ and

$$0 < 2sig(\hat{x}\hat{y}) - sig(\hat{d}) < 2^{-3}.$$

Thus,

$$sig(\hat{x}\hat{y}) < \frac{1}{2}sig(\hat{d}) + 2^{-4} \leq 1 + 2^{-4}$$

and

$$sig(\hat{d}) > 2sig(\hat{x}\hat{y}) - 2^{-3} \geq 2 - 2^{-3}.$$

It follows that

$$add < 2^{P-1}(2 + 1 + 2^{-4} - 2 + 2^{-3}) < 2^{P-1} + 2^{P-2}.$$

But $add > 2^{P-1}(2 + 1 - 2) = 2^{P-1}$, hence $\rho < 2^{P-2}$ and $\rho[P-2] = 0$.

Case 3: $expo(\hat{x}\hat{y}) = expo(\hat{d}) - 1$

In this case, $|\hat{x}\hat{y}| < |\hat{d}|$ and

$$0 < sig(\hat{d}) - \frac{1}{2}sig(\hat{x}\hat{y}) < 2^{-3}.$$

Thus, $sig(\hat{d}) < 1 + 2^{-3}$, $sig(\hat{x}\hat{y}) > 2 - 2^{-2}$, and

$$add > 2^{P-1}(2 + 2 - 2^{-2} - 1 - 2^{-3}) > 3 \cdot 2^{P-1} - 2^{P-2} = 2 \cdot 2^{P-1} + 2^{P-2}.$$

But

$$add < 2^{P-1}(2 + 2 - 1) = 3 \cdot 2^{P-1},$$

hence $\rho > 2^{P-2}$ and $\rho[P-2] = 1$.

(b) Note that by Lemmas 3.1 and 3.2, *inexact* $= 0$ iff $\hat{x}\hat{y}$ is M-exact. Thus, if $\hat{x}\hat{y} = \hat{d}$, then *inexact* $= 0$ and $add = 2^P$, which implies $\rho = 0$, and hence $get\text{-}man(z)[M-2:0] = 0$.

Conversely, suppose

$$get\text{-}man(z)[M-2:0] = \rho[P-2:P-M] = inexact = 0.$$

Then $sig(\hat{x}\hat{y})$ is M-exact, i.e., $2^{M-1}sig(\hat{x}\hat{y}) \in \mathbb{Z}$, hence $2^{P-1}sig(\hat{x}\hat{y})$ is divisible by 2^{P-M}. Similarly, $2^{P-1}sig(\hat{d})$ is divisible by 2^{P-M}, and hence, so are add and ρ. Thus,

$$\rho = (\rho/2^{P-M})2^{P-M} = \lfloor \rho/2^{P-M} \rfloor 2^{P-M} = \rho[P-2:P-M]2^{P-M} = 0.$$

Since $\hat{x}\hat{y} = -\hat{d}$ is impossible, we need only show $|\hat{x}\hat{y}| = |\hat{d}|$. In view of (a), we may assume $|\hat{x}\hat{y}| \geq |\hat{d}|$. Thus, there are two cases to consider:

Case 1: $expo(\hat{x}\hat{y}) = expo(\hat{d})$

In this case, $sig(\hat{x}\hat{y}) \geq sig(\hat{d})$, which implies

$$\rho = 2^{P-1}(sig(\hat{x}\hat{y}) - sig(\hat{d})) = 0,$$

hence $sig(\hat{x}\hat{y}) = sig(\hat{d})$ and $|\hat{x}\hat{y}| = |\hat{d}|$.

Case 2: $expo(\hat{x}\hat{y}) = expo(\hat{d}) + 1$

If this were to occur, then we would have

$$\rho = 2^{P-1}(1 + sig(\hat{x}\hat{y}) - sig(\hat{d})) = 0,$$

implying $sig(\hat{d}) = 1 + sig(\hat{x}\hat{y}) \geq 2$, which is impossible. □

4 Division and Square Root

4.1 The Program *FPU-DIV-SQRT*

The hardware for division and square root is represented by the program *FPU-DIV-SQRT*, shown in Figures 3 and 4. Our analysis will be based on an execution of

$$\textit{FPU-DIV-SQRT(op,pc,rc,a,b,z)},$$

with inputs as follows:

(a) $op \in \{\texttt{OP-DIV}, \texttt{OP-SQRT}\}$;

(b) pc is an external precision control specifier;

(c) rc is a rounding control specifier;

(d) a and b are normal encodings.

In the case $op = \texttt{OP-DIV}$, the output z represents an appropriately rounded approximation of the quotient $\hat{a}/\hat{b}$; when $op = \texttt{OP-SQRT}$, a is ignored and an approximation of $\sqrt{\hat{b}}$ is returned.

Both operations are based on Goldschmidt's Algorithm [2], a variant of Newton-Raphson approximation. Our analysis of division will involve a sequence $\xi_0, \xi_1, \xi_2, \xi_3$ of approximations to $1/\hat{b}$, where ξ_0 is derived from a table and the other ξ_i are computed by three successive Newton-Raphson iterations. The square root involves a similar sequence of approximations to $1/\sqrt{\hat{b}}$.

Although the algorithm does not explicitly compute the ξ_i for $i > 0$, a sequence of calls to *FPU-MUL* produces an encoding q of either $\hat{a}\xi_i$ or $\hat{b}\xi_i$, modulo rounding error, according to whether $op = \texttt{OP-DIV}$ or $op = \texttt{OP-SQRT}$, where (a) $i = 1$ if $pc = \texttt{PC-32}$, (b) $i = 2$ if $pc = \texttt{PC-64}$, and (c) $i = 3$ if $pc = \texttt{PC-80}$ or $pc = \texttt{PC-87}$. Lemmas 4.9 and 4.13 give estimates of the errors $|\hat{q} - \hat{a}/\hat{b}|$ and $|\hat{q} - \sqrt{\hat{b}}|$. Note that the constraint $M \geq 75$ on the multiplier width is required in the proofs of these lemmas.

The approximation $\hat{q}$ is compared to the exact value by means of a final call to *FPU-MUL* with $op = \texttt{OP-BACK}$. Using the results of this comparison, q is then adjusted to produce the correctly rounded result z. The correctness of this result is guaranteed by Theorems 2 and 3.

4.2 Initial Approximation

The initial approximation x_0 to the reciprocal of b, in the case $op = \texttt{OP-DIV}$, is derived from a pair of tables, each consisting of 2^{10} bit vectors, which we represent by the functions *recip-rom-p* and *recip-rom-n*. If $sig(\hat{b})$ has the binary representation $1.b_1b_2b_3\ldots$, then the bit vectors

$$b_1b_2\ldots b_9b_{10} = \textit{get-man}(b)[M-2:M-11]$$

and

$$b_1\ldots b_5b_{11}\ldots b_{15} = \textit{cat}(\textit{get-man}(b)[M-2:M-6], \textit{get-man}(b)[M-12:M-16], 5)$$

are used as indices into these tables. The results are added and the 16-bit sum is appended to a leading 1 and $M-17$ trailing 0's to produce $\textit{get-man}(x_0)$. For $op =$

```
Program FPU-DIV-SQRT(op,pc,rc,a,b,z):

if $op =$ OP-DIV then
  {$sign \leftarrow get\text{-}sign(a)$ ^ $get\text{-}sign(b)$;
   $p\text{-}value \leftarrow recip\text{-}rom\text{-}p(get\text{-}man(b)[M-2 : M-11])$;
   $n\text{-}value \leftarrow recip\text{-}rom\text{-}n(cat(get\text{-}man(b)[M-2 : M-6],$
                                  $get\text{-}man(b)[M-12 : M-16],$
                                  $5))$;
   $estimate \leftarrow (p\text{-}value + n\text{-}value)[16 : 0]$;
   $x_0 \leftarrow (get\text{-}sign(b),$
         $2^{M-17}estimate \mid 2^{M-1},$
         $(2^{18} - 2 + comp1(get\text{-}expo(b), 18) + estimate[16])[17 : 0])$;
   $FPU\text{-}MUL($OP-DIV, PC-*, NIL, RC-NEAR, $b, x_0, d_0, r_0,$ NIL, NIL);
   $FPU\text{-}MUL($OP-MUL, PC-*, NIL, RC-NEAR, $a, x_0, n_0,$ NIL, NIL, NIL);
   if $pc =$ PC-32
     then $FPU\text{-}MUL($OP-LAST, PC-*, $pc$, RC-NEAR, $n_0, r_0, q,$ NIL, NIL, NIL)
     else {$FPU\text{-}MUL($OP-DIV, PC-*, NIL, RC-NEAR, $d_0, r_0, d_1, r_1,$ NIL, NIL);
          $FPU\text{-}MUL($OP-MUL, PC-*, NIL, RC-NEAR, $n_0, r_0, n_1,$ NIL, NIL, NIL);
          if $pc =$ PC-64
            then $FPU\text{-}MUL($OP-LAST, PC-*, $pc$, RC-NEAR, $n_1, r_1, q,$ NIL, NIL, NIL)
            else {$FPU\text{-}MUL($OP-DIV, PC-*, NIL, RC-NEAR, $d_1, r_1, d_2, r_2,$ NIL, NIL);
                 $FPU\text{-}MUL($OP-MUL, PC-*, NIL, RC-NEAR, $n_1, r_1, n_2,$ NIL, NIL, NIL);
                 $FPU\text{-}MUL($OP-LAST, PC-*, $pc$, RC-NEAR, $n_2, r_2, q,$ NIL, NIL, NIL)}};
  $FPU\text{-}MUL($OP-BACK, PC-*, NIL, RC-CHOP, $b, q, rem,$ NIL, $a, inexact)$}

else if $op =$ OP-SQRT then
  {$sign \leftarrow 0$;
   $p\text{-}value \leftarrow sqrt\text{-}rom\text{-}p(cat(get\text{-}expo(b)[0], get\text{-}man(b)[M-2 : M-11], 10))$;
   $n\text{-}value \leftarrow sqrt\text{-}rom\text{-}n(cat(get\text{-}expo(b)[0],$
                               $cat(get\text{-}man(b)[M-2 : M-6],$
                                   $get\text{-}man(b)[M-12 : M-16],$
                                   $5),$
                               $10))$;
   $estimate \leftarrow (p\text{-}value + n\text{-}value)[16 : 0]$;
   $x_0 \leftarrow (get\text{-}sign(b),$
         $2^{M-17}estimate \mid 2^{M-1},$
         $shr((2^{18} + 2^{17} - 3 + comp1(get\text{-}expo(b), 19) + estimate[16])[18 : 0], 0, 19))$;
   $FPU\text{-}MUL($OP-MUL, PC-*, NIL, RC-NEAR, $x_0, x_0, t_0,$ NIL, NIL, NIL);
   $FPU\text{-}MUL($OP-MUL, PC-*, NIL, RC-NEAR, $b, x_0, d_0,$ NIL, NIL, NIL);
   $FPU\text{-}MUL($OP-SQRT, PC-*, NIL, RC-NEAR, $b, t_0, n_0, r_0,$ NIL, NIL);
```

Figure 3: *FPU-DIV-SQRT*

```
  if $pc = \texttt{PC-32}$
    then $FPU\text{-}MUL(\texttt{OP-LAST}, \texttt{PC-*}, pc, \texttt{RC-NEAR}, d_0, r_0, q, \texttt{NIL}, \texttt{NIL}, \texttt{NIL})$
    else $\{FPU\text{-}MUL(\texttt{OP-MUL}, \texttt{PC-*}, \texttt{NIL}, \texttt{RC-NEAR}, r_0, r_0, t_1, \texttt{NIL}, \texttt{NIL}, \texttt{NIL});$
          $FPU\text{-}MUL(\texttt{OP-MUL}, \texttt{PC-*}, \texttt{NIL}, \texttt{RC-NEAR}, d_0, r_0, d_1, \texttt{NIL}, \texttt{NIL}, \texttt{NIL});$
          $FPU\text{-}MUL(\texttt{OP-SQRT}, \texttt{PC-*}, \texttt{NIL}, \texttt{RC-NEAR}, n_0, t_1, n_1, r_1, \texttt{NIL}, \texttt{NIL});$
          if $pc = \texttt{PC-64}$
            then $FPU\text{-}MUL(\texttt{OP-LAST}, \texttt{PC-*}, pc, \texttt{RC-NEAR}, d_1, r_1, q, \texttt{NIL}, \texttt{NIL}, \texttt{NIL})$
            else $\{FPU\text{-}MUL(\texttt{OP-MUL}, \texttt{PC-*}, \texttt{NIL}, \texttt{RC-NEAR}, r_1, r_1, t_2, \texttt{NIL}, \texttt{NIL}, \texttt{NIL});$
                  $FPU\text{-}MUL(\texttt{OP-MUL}, \texttt{PC-*}, \texttt{NIL}, \texttt{RC-NEAR}, d_1, r_1, d_2, \texttt{NIL}, \texttt{NIL}, \texttt{NIL});$
                  $FPU\text{-}MUL(\texttt{OP-SQRT}, \texttt{PC-*}, \texttt{NIL}, \texttt{RC-NEAR}, n_1, t_2, n_2, r_2, \texttt{NIL}, \texttt{NIL});$
                  $FPU\text{-}MUL(\texttt{OP-LAST}, \texttt{PC-*}, pc, \texttt{RC-NEAR}, d_2, r_2, q, \texttt{NIL}, \texttt{NIL}, \texttt{NIL})\}\};$
 $FPU\text{-}MUL(\texttt{OP-BACK}, \texttt{PC-*}, \texttt{NIL}, \texttt{RC-CHOP}, q, q, rem, \texttt{NIL}, b, inexact)\};$

if $get\text{-}man(rem)[M-2:0] = 0$
  then $rem\text{-}zero \leftarrow comp1(inexact, 1)$
  else $rem\text{-}zero \leftarrow 0;$
$rem\text{-}neg \leftarrow comp1(get\text{-}man(rem)[M-2], 1)\ \&comp1(rem\text{-}zero, 1);$
$rem\text{-}pos \leftarrow get\text{-}man(rem)[M-2];$
$q\text{-}lsb \leftarrow get\text{-}man(q)[M - mbits(pc)];$
$q\text{-}guard \leftarrow get\text{-}man(q)[M - mbits(pc) - 1];$
if $op = \texttt{OP-DIV} \wedge get\text{-}man(a) = 0$ then
  $z \leftarrow (sign, 0, get\text{-}expo(a))$
else if $op = \texttt{OP-SQRT} \wedge get\text{-}man(b) = 0$ then
  $z \leftarrow (sign, 0, get\text{-}expo(b))$
else if $((rc = \texttt{RC-POS} \wedge sign = 1) \vee (rc = \texttt{RC-NEG} \wedge sign = 0) \vee rc = \texttt{RC-CHOP})$
        $\wedge q\text{-}guard = 0 \wedge rem\text{-}neg = 1$ then
  if $get\text{-}man(q)\ \&\ (2^M - 2^{M-mbits(pc)}) = 2^{M-1}$
    then $z \leftarrow (sign, 2^M - 2^{M-mbits(pc)}, dec1(get\text{-}expo(q), 18))$
    else $z \leftarrow (sign,$
               $((get\text{-}man(q)\ \&\ (2^M - 2^{M-mbits(pc)})) + 2^M - 2^{M-mbits(pc)})[M-1:0],$
               $get\text{-}expo(q))$
else if $(((rc = \texttt{RC-POS} \wedge sign = 0) \vee (rc = \texttt{RC-NEG} \wedge sign = 1))$
         $\wedge(q\text{-}guard = 1 \vee rem\text{-}pos = 1))$
        $\vee(rc = \texttt{RC-NEAR} \wedge q\text{-}guard = 1 \wedge rem\text{-}pos = 1)$
        $\vee(rc = \texttt{RC-NEAR} \wedge q\text{-}guard = 1 \wedge rem\text{-}zero = 1 \wedge q\text{-}lsb = 1)$ then
  if $get\text{-}man(q)\ \&\ (2^M - 2^{M-mbits(pc)}) = 2^M - 2^{M-mbits(pc)}$
    then $z \leftarrow (sign, 2^{M-1}, (get\text{-}expo(q) + 1)[17:0])$
    else $z \leftarrow (sign,$
               $((get\text{-}man(q)\ \&\ (2^M - 2^{M-mbits(pc)})) + 2^{M-mbits(pc)})[M-1:0],$
               $get\text{-}expo(q))$
else $z \leftarrow (sign, get\text{-}man(q)\ \&\ (2^M - 2^{M-mbits(pc)}), get\text{-}expo(q)).$
```

Figure 4: *FPU-DIV-SQRT (continued)*

`OP-SQRT`, a separate pair of tables, represented by the functions *sqrt-rom-p* and *sqrt-rom-n*, is similarly used to derive an initial approximation to the reciprocal of the square root of b.

The functions R_0, S_0, and S_1, which are defined in terms of these functions, represent the computation of $\mathit{get\text{-}man}(x_0)$ in the three cases listed in Lemma 4.4 below.

Definition 4.1 *For all* $i \in \mathbb{N}$,

(a) $R_0(i) = 2^{16} + \mathit{recip\text{-}rom\text{-}p}(i[14:5]) + \mathit{recip\text{-}rom\text{-}n}(\mathit{cat}(i[14:10], i[4:0], 5))$;
(b) $S_0(i) = 2^{16} + \mathit{sqrt\text{-}rom\text{-}p}(i[14:5]) + \mathit{sqrt\text{-}rom\text{-}n}(\mathit{cat}(i[14:10], i[4:0], 5))$;
(c) $S_1(i) = 2^{16} + \mathit{sqrt\text{-}rom\text{-}p}(2^{10} + i[14:5])$
$+\mathit{sqrt\text{-}rom\text{-}n}(2^{10} + \mathit{cat}(i[14:10], i[4:0], 5))$.

While space does not allow a complete listing of the tables here, we list instead the following three lemmas, which contain all required relevant information, and which have all been verified by direct computation, using ACL2:

Lemma 4.1 *For all* $i \in \mathbb{N}$, *if* $i < 2^{15}$, *then* $R_0(i) \in \mathbb{N}$, $S_0(i) \in \mathbb{N}$, $S_1(i) \in \mathbb{N}$, *and*

$$\mathit{expo}(R_0(i)) = \mathit{expo}(S_0(i)) = \mathit{expo}(S_1(i)) = 16.$$

Lemma 4.2 *For all* $i \in \mathbb{N}$, *if* $i < 2^{15}$, *then*

(a) $2^{32} - 3 \cdot 2^{16} < R_0(i)(2^{15} + i) < R_0(i)(2^{15} + i + 1) < 2^{32} + 3 \cdot 2^{16}$;
(b) $2^{48} - 3 \cdot 2^{32} < S_0(i)^2(2^{15} + i) < S_0(i)^2(2^{15} + i + 1) < 2^{48} + 3 \cdot 2^{32}$;
(c) $2^{49} - 3 \cdot 2^{33} < S_1(i)^2(2^{15} + i) < S_1(i)^2(2^{15} + i + 1) < 2^{49} + 3 \cdot 2^{33}$;

Lemma 4.3 *For all* $i \in \mathbb{N}$, *if* $i < 2^{15}$, *then* $S_0(i)^2 < 2^{33} \leq S_1(i)^2$.

The relationship between x_0 and b may be described in terms of R_0, S_0, and S_1:

Lemma 4.4 *Let* $I = \mathit{get\text{-}man}(b)[M-2:M-16]$. *Assume that if* $\mathit{op} =$ `OP-DIV`, *then* $\mathit{get\text{-}expo}(b) \leq 2^{18} - 3$. *Then* x_0 *is normal and*

$$\text{(a) } \mathit{sgn}(\hat{x_0}) = \begin{cases} \mathit{sgn}(\hat{b}) & \text{if } \mathit{op} = \texttt{OP-DIV} \\ 1 & \text{if } \mathit{op} = \texttt{OP-SQRT}; \end{cases}$$

$$\text{(b) } \mathit{sig}(\hat{x_0}) = \begin{cases} 2^{-16}R_0(I) & \text{if } \mathit{op} = \texttt{OP-DIV} \\ 2^{-16}S_0(I) & \text{if } \mathit{op} = \texttt{OP-SQRT} \text{ and } \mathit{get\text{-}expo}(b)[0] = 0 \\ 2^{-16}S_1(I) & \text{if } \mathit{op} = \texttt{OP-SQRT} \text{ and } \mathit{get\text{-}expo}(b)[0] = 1; \end{cases}$$

$$\text{(c) } \mathit{expo}(\hat{x_0}) = \begin{cases} -\mathit{expo}(\hat{b}) - 1 & \text{if } \mathit{op} = \texttt{OP-DIV} \\ -\lfloor \mathit{expo}(\hat{b})/2 \rfloor - 1 & \text{if } \mathit{op} = \texttt{OP-SQRT}. \end{cases}$$

Proof: First consider the case $\mathit{op} =$ `OP-DIV`. By Lemma 2.5,

$$\mathit{get\text{-}man}(b)[M-2:M-11] = \mathit{get\text{-}man}(b)[M-2:M-16][14:5] = I[14:5],$$

hence $\mathit{p\text{-}value} = \mathit{recip\text{-}rom\text{-}p}(I[14:5])$. Similarly,

$$\mathit{n\text{-}value} = \mathit{recip\text{-}rom\text{-}n}(\mathit{cat}(I[14:10], I[4:0], 5)).$$

By Lemma 4.1,

$$\textit{p-value} + \textit{n-value} = R_0(I) - 2^{16} < 2^{17} - 2^{16} = 2^{16},$$

hence

$$\textit{estimate} = \textit{p-value} + \textit{n-value} < 2^{16}$$

and by Lemma 2.8,

$$\begin{aligned} \textit{get-man}(x_0) &= 2^{M-17}\textit{estimate} \mid 2^{M-1} = 2^{M-17}(\textit{estimate} \mid 2^{16}) \\ &= 2^{M-17}(\textit{estimate} + 2^{16}) = 2^{M-17}R_0(I). \end{aligned}$$

Since $\textit{estimate}[16] = 0$ and $\textit{get-expo}(b) \leq 2^{18} - 3$,

$$\textit{get-expo}(x_0) = \textit{rem}(2^{18} - 2 + 2^{18} - \textit{get-expo}(b) - 1, 2^{18}) = 2^{18} - 3 - \textit{get-expo}(b).$$

The `OP-DIV` case now follows easily from Lemmas 4.1 and 2.15.

In the case $op =$ `OP-SQRT`, we may similarly show that $\textit{get-man}(x_0) = 2^{M-17}S_j(I)$, where $j = \textit{get-expo}(b)[0]$. Now

$$\begin{aligned} &(2^{18} + 2^{17} - 3 + \textit{compl}(\textit{get-expo}(b), 19) + \textit{estimate}[16])[18:0] \\ &= (2^{18} + 2^{17} - 3 + \textit{compl}(\textit{get-expo}(b), 19))[18:0] \\ &= \textit{rem}(2^{18} + 2^{17} - 3 + \textit{compl}(\textit{get-expo}(b), 19), 2^{19}) \\ &= \textit{rem}(2^{18} + 2^{17} - 3 + 2^{19} - \textit{get-expo}(b) - 1, 2^{19}) \\ &= \textit{rem}(2^{18} + 2^{17} - 3 + 2^{19} - (\textit{expo}(\hat{b}) + 2^{17} - 1) - 1, 2^{19}) \\ &= \textit{rem}(2^{18} - \textit{expo}(\hat{b}) - 3, 2^{19}) \\ &= 2^{18} - \textit{expo}(\hat{b}) - 3. \end{aligned}$$

Thus,

$$\begin{aligned} \textit{get-expo}(x_0) &= \textit{shr}(2^{18} - \textit{expo}(\hat{b}) - 3, 0, 19) \\ &= \lfloor (2^{18} - \textit{expo}(\hat{b}) - 3)/2 \rfloor \\ &= 2^{17} - 1 + \lfloor -(\textit{expo}(\hat{b}) + 1)/2 \rfloor, \end{aligned}$$

and

$$\textit{expo}(\hat{x_0}) = \lfloor -(\textit{expo}(\hat{b}) + 1)/2 \rfloor = -\lfloor \textit{expo}(\hat{b})/2 \rfloor - 1. \Box$$

The error associated with x_0 is characterized by the next two lemmas, which also establish the bounds required by Lemma 3.5:

Lemma 4.5 *If $op =$ `OP-DIV` and $\textit{get-expo}(b) \leq 2^{18} - 3$, then*
(a) $|1 - \hat{x}_0 \hat{b}| < 3 \cdot 2^{-16}$;   (b) $3/2 < \textit{sig}(\hat{x_0})\textit{sig}(\hat{b}) < 3$.

Proof: (a) By Lemma 4.4,

$$\hat{x_0}\hat{b} = \textit{sig}(\hat{x_0})\textit{sig}(\hat{b})2^{\textit{expo}(\hat{x_0}) + \textit{expo}(\hat{b})} = \textit{sig}(\hat{x_0})\textit{sig}(\hat{b})/2.$$

Let $I = \textit{get-man}(b)[M-2:M-16]$. Since $2^{M-1} \leq \textit{get-man}(b) < 2^M$,

$$\begin{aligned} I &= \lfloor \textit{rem}(\textit{get-man}(b), 2^{M-1})/2^{M-16} \rfloor = \lfloor (\textit{get-man}(b) - 2^{M-1})/2^{M-16} \rfloor \\ &= \lfloor \textit{get-man}(b)/2^{M-16} - 2^{15} \rfloor, \end{aligned}$$

hence

$$get\text{-}man(b)/2^{M-16} - 2^{15} - 1 < I \leq get\text{-}man(b)/2^{M-16} - 2^{15},$$

which along with Lemma 2.15, implies

$$2^{-15}(2^{15} + I) \leq sig(\hat{b}) < 2^{-15}(2^{15} + I + 1).$$

Thus, by Lemmas 4.4 and 4.2,

$$1 - 3 \cdot 2^{-16} < 2^{-32}R_0(I)(2^{15} + I) \leq \hat{x_0}\hat{b} < 2^{-32}R_0(I)(2^{15} + I + 1) < 1 + 3 \cdot 2^{-16}.$$

(b) This follows from (a) and the observation that $sig(\hat{x_0})sig(\hat{b}) = 2\hat{x_0}\hat{b}$. □

Lemma 4.6 *If* $op =$ `OP-SQRT`*,* $\hat{b} > 0$*, and* $get\text{-}expo(b) \leq 2^{18} - 3$*, then*
(a) $|1 - \hat{x_0}^2\hat{b}| < 3 \cdot 2^{-16}$*;*
(b) $3/2 < sig(\hat{x_0}^2)sig(\hat{b}) < 3$*;*
(c) $\hat{x_0}^2$ *is representable.*

Proof: Let $I = get\text{-}man(b)[M-2 : M-16]$ and $expo(\hat{b}) = 2r + s$, where $0 \leq s \leq 1$.

Case 1: $s = 0$

(a) In this case, $get\text{-}expo(b)[0] = 1$. By Lemma 4.4,

$$\begin{aligned}\hat{x_0}^2\hat{b} &= sig(\hat{x_0})^2 sig(\hat{b})2^{2expo(\hat{x_0})+expo(\hat{b})} = sig(\hat{x_0})^2 sig(\hat{b})2^{2(-r-1)+2r} \\ &= sig(\hat{x_0})^2 sig(\hat{b})/4 = 2^{-34}S_1(I)^2 sig(\hat{b}).\end{aligned}$$

Thus, by Lemma 4.2,

$$1 - 3 \cdot 2^{-16} < 2^{-49}S_1(I)^2(2^{15} + I) \leq \hat{x_0}^2\hat{b} < 2^{-49}S_1(I)^2(2^{15} + I + 1) < 1 + 3 \cdot 2^{-16}.$$

(b) By Lemmas 4.4 and 4.3, $sig(\hat{x_0})^2 = 2^{-32}S_1(I)^2 \geq 2$, which implies $sig(\hat{x_0}^2) = sig(\hat{x_0})^2/2$. Thus,

$$\hat{x_0}^2\hat{b} = sig(\hat{x_0})^2 sig(\hat{b})/4 = sig(\hat{x_0}^2)sig(\hat{b})/2.$$

The claim now follows from (a).

(c) By Lemmas 4.1 and 4.4, $\hat{x_0}$ is 17-exact, and it follows that $\hat{x_0}^2$ is M-exact. Since $expo(\hat{b}) \geq 1 - 2^{17}$,

$$expo(\hat{x_0}) \leq -\lfloor(1 - 2^{17})/2\rfloor - 1 = 2^{16} - 1$$

and

$$expo(\hat{x_0}^2) \leq 2expo(\hat{x_0}) + 1 \leq 2^{17} - 1.$$

But since $expo(\hat{b}) = get\text{-}expo(b) - (2^{17} - 1) \leq (2^{18} - 3) - (2^{17} - 1) = 2^{17} - 2$,

$$\hat{x_0}^2 = sig(\hat{x_0}^2)sig(\hat{b})/2\hat{b} \geq sig(\hat{b})/2\hat{b} = 2^{-1-expo(\hat{b})} \geq 2^{1-2^{17}},$$

hence $expo(\hat{x_0}^2) \geq 1 - 2^{17}$.

Case 2: $s = 1$

(a) In this case, $get\text{-}expo(b)[0] = 0$. By Lemma 4.4,

$$\begin{aligned}\hat{x_0}^2\hat{b} &= sig(\hat{x_0})^2 sig(\hat{b})2^{2expo(\hat{x_0})+expo(\hat{b})} = sig(\hat{x_0})^2 sig(\hat{b})2^{2(-r-1)+2r+1} \\ &= sig(\hat{x_0})^2 sig(\hat{b})/2 = 2^{-33}S_0(I)^2 sig(\hat{b}).\end{aligned}$$

Thus, by Lemma 4.2,

$$1 - 3 \cdot 2^{-16} < 2^{-48} S_0(I)^2 (2^{15} + I) \leq \hat{x_0}^2 \hat{b} < 2^{-48} S_0(I)^2 (2^{15} + I + 1) < 1 + 3 \cdot 2^{-16}.$$

(b) By Lemmas 4.4 and 4.3, $sig(\hat{x_0})^2 = 2^{-32} S_0(I)^2 < 2$, which implies $sig(\hat{x_0}^2) = sig(\hat{x_0})^2$. Thus,

$$\hat{x_0}^2 \hat{b} = sig(\hat{x_0})^2 sig(\hat{b})/2 = sig(\hat{x_0}^2) sig(\hat{b})/2. \Box$$

(c) As in Case 1, $\hat{x_0}^2$ is M-exact and $expo(\hat{x_0}^2) \leq 2^{17} - 1$. Since $expo(\hat{b}) \leq 2^{17} - 2$ and $expo(\hat{b})$ is odd, $expo(\hat{b}) \leq 2^{17} - 3$, hence

$$expo(\hat{x_0}) \geq -\lfloor (2^{17} - 3)/2 \rfloor - 1 = 1 - 2^{16}$$

and

$$expo(\hat{x_0}^2) \geq 2 expo(\hat{x_0}) \geq 2 - 2^{17}.$$

4.3 The Operation `OP-DIV`

Given an initial approximation ξ_0 of $1/\hat{b}$, the Newton-Raphson formula

$$\xi_i = \xi_{i-1}(2 - \hat{b}\xi_{i-1})$$

gives a converging sequence of approximations $\xi_1, \xi_2, \ldots$ The relative error of ξ_i is

$$\left| \frac{1/\hat{b} - \xi_i}{1/\hat{b}} \right| = |1 - \hat{b}\xi_i|.$$

Thus, the following lemma (which is proved by simple arithmetic) shows that this sequence is quadratically convergent:

Lemma 4.7 *Let $b, x \in \mathbb{Q}$ and let $y = x(2 - bx)$. Then $1 - by = (1 - bx)^2$.*

Using Lemma 4.7, we shall derive an error estimate for $\hat{q}$ as an approximation of $\hat{a}/\hat{b}$. First, we prove the following technical lemma:

Lemma 4.8 *Assume $\hat{q}$ is $(\mu + 1)$-exact, where $\mu \geq 1$, and $\hat{q} \neq 0$. Let $\zeta \in \mathbb{Q}$ satisfy*

$$expo(\zeta) \leq expo(\hat{q}),$$

$$|\hat{q} - \zeta| \leq 2^{expo(\zeta) - \mu - 1},$$

and

$$|\hat{a}/\hat{b} - \zeta| < 2^{expo(\hat{a}/\hat{b}) - \mu - 2}.$$

Then

$$|\hat{q} - \hat{a}/\hat{b}| < 2^{min(expo(\hat{q}), expo(\hat{a}/\hat{b})) - \mu}.$$

Proof: First note that $|\hat{q}| \geq \frac{3}{4}|\zeta| > \frac{9}{16}|\hat{a}/\hat{b}|$, hence $expo(\hat{q}) \geq expo(\hat{a}/\hat{b}) - 1$. Since

$$|\hat{q} - \hat{a}/\hat{b}| \leq |\hat{q} - \zeta| + |\hat{a}/\hat{b} - \zeta| < 2^{expo(\hat{q}) - \mu - 1} + 2^{expo(\hat{q}) - \mu - 1} = 2^{expo(\hat{q}) - \mu},$$

we may assume $expo(\hat{a}/\hat{b}) < expo(\hat{q})$. But $|\hat{a}/\hat{b}| > |\hat{q}|/2$, hence $expo(\hat{a}/\hat{b}) = expo(\hat{q}) - 1$. We may also assume $expo(\zeta) = expo(\hat{q})$, for otherwise $expo(\zeta) \leq expo(\hat{a}/\hat{b})$ and

$$|\hat{q} - \hat{a}/\hat{b}| \leq |\hat{q} - \zeta| + |\hat{a}/\hat{b} - \zeta| < 2^{expo(\zeta) - \mu - 1} + 2^{expo(\hat{a}/\hat{b}) - \mu - 1} \leq 2^{expo(\hat{a}/\hat{b}) - \mu}.$$

If $|\hat{q}| > 2^{expo(\hat{q})}$, then $|\hat{q}| \geq 2^{expo(\hat{q})} + 2^{expo(\hat{q})-\mu}$ by Lemma 2.13, and

$$|\hat{q} - \hat{a}/\hat{b}| \geq |\hat{q}| - |\hat{a}/\hat{b}| > 2^{expo(\hat{q})} + 2^{expo(\hat{q})-\mu} - 2^{expo(\hat{a}/\hat{b})+1} = 2^{expo(\hat{q})-\mu}.$$

Therefore, $|\hat{q}| = 2^{expo(\hat{q})}$, which implies $|\zeta| \geq |\hat{q}|$ and

$$|q - \hat{a}/\hat{b}| = |\hat{q}| - |\hat{a}/\hat{b}| \leq |\zeta| - |\hat{a}/\hat{b}| \leq |\zeta - \hat{a}/\hat{b}| < 2^{expo(\hat{a}/\hat{b})-\mu}.\Box$$

We shall assume here that $\hat{a}$ and $\hat{b}$ are both positive; this assumption will be relieved in the proof of the main theorem:

Lemma 4.9 *Assume $op =$ `OP-DIV`, $\hat{a} > 0$, $\hat{b} > 0$, $expo(\hat{b}) \leq 2^{17} - 2$, $3 \cdot 2^{-2^{17}} < |\hat{a}/\hat{b}| < 3 \cdot 2^{2^{17}-1}$, and $mbits(pc) = \mu$. Then q is normal, $\hat{q}$ is $(\mu + 1)$-exact and*

$$|\hat{q} - \hat{a}/\hat{b}| < 2^{min(expo(\hat{q}),expo(\hat{a}/\hat{b}))-\mu}.$$

Proof: Let $\alpha = 2^{-M}$, $\beta = 2^{expo(\hat{a}/\hat{b})}$, and $\epsilon = 3/2^{16}$. We define a sequence of approximations ξ_i of $\hat{a}/\hat{b}$ by

$$\xi_i = \begin{cases} \hat{x_0} & \text{if } i = 0 \\ \xi_{i-1}(2 - \hat{b}\xi_{i-1}) & \text{if } i > 0. \end{cases}$$

Since $\hat{a}$ and $\hat{b}$ are positive, so are the ξ_i, as well as every product computed by *FPU-MUL*. By Lemmas 4.5 and 4.7, $|1 - \hat{b}\xi_i| < \epsilon^{2^i}$ for all i. Thus, $\hat{b}\xi_i < 1 + \epsilon^{2^i}$ and $2 - \hat{b}\xi_i < 1 + \epsilon^{2^i}$. We also have

$$\hat{a}\xi_i = (\hat{a}/\hat{b})(\hat{b}\xi_i) < 2\beta(1 + \epsilon^{2^i})$$

and

$$|\hat{a}/\hat{b} - \hat{a}\xi_i| = (\hat{a}/\hat{b})|1 - \hat{b}\xi_i| < (\hat{a}/\hat{b})\epsilon^{2^i} < 2\beta\epsilon^{2^i}.$$

By Theorem 1, $\hat{d_0} = near(\hat{b}\hat{x_0}, M) = near(\hat{b}\xi_0, M)$, hence by Lemma 2.26,

$$|\hat{d_0} - \hat{b}\xi_0| \leq 2^{expo(\hat{b}\xi_0)-M} \leq 2^{-M} = \alpha.$$

Note that our bounds for $|\hat{a}/\hat{b}|$ ensure that the hypotheses of Theorem 1 are satisfied by $x = a$ and $y = x_0$. Thus,

$$|\hat{n_0} - \hat{a}\xi_0| \leq 2^{expo(\hat{a}\xi_0)-M} \leq 2^{expo(\hat{a}/\hat{b})+1-M} = 2\alpha\beta,$$

and by Lemma 3.5 (the hypotheses of which are ensured by Lemma 4.5),

$$0 < 2 - \hat{b}\xi_0 - 2\alpha \leq \hat{r_0} < 2 - \hat{b}\xi_0.$$

Therefore,

$$\begin{aligned} \hat{n_0}\hat{r_0} &< (\hat{a}\xi_0 + 2\alpha\beta)(2 - \hat{b}\xi_0) = \hat{a}\xi_1 + 2\alpha\beta(2 - \hat{b}\xi_0) < \hat{a}\xi_1 + 2\alpha\beta(1 + \epsilon) \\ &< \hat{a}\xi_1 + 2\alpha\beta + 2^{-13}\alpha\beta, \end{aligned}$$

$$\begin{aligned} \hat{n_0}\hat{r_0} &\geq (\hat{a}\xi_0 - 2\alpha\beta)(2 - \hat{b}\xi_0 - 2\alpha) = \hat{a}\xi_1 - 2\alpha\beta(2 - \hat{b}\xi_0) - 2\alpha\hat{a}\xi_0 + 4\alpha^2\beta \\ &> \hat{a}\xi_1 - 2\alpha\beta(1 + \epsilon) - 2\alpha 2\beta(1 + \epsilon) > \hat{a}\xi_1 - 6\alpha\beta - 2^{-12}\alpha\beta, \end{aligned}$$

and

$$\begin{aligned}|\hat{n_0}\hat{r_0} - \hat{a}/\hat{b}| &\leq |\hat{n_0}\hat{r_0} - \hat{a}\xi_1| + |\hat{a}\xi_1 - \hat{a}/\hat{b}| < 7\alpha\beta + 2\beta\epsilon^2 \\ &< (7 \cdot 2^{-75} + 9 \cdot 2^{-31})\beta < 2^{-27}\beta \\ &= 2^{expo(\hat{a}/\hat{b})-27}.\end{aligned}$$

Suppose $pc =$ `PC-32`. Then $\mu = 24$ and

$$|\hat{n_0}\hat{r_0} - \hat{a}/\hat{b}| < 2^{expo(\hat{a}/\hat{b})-27} < 2^{expo(\hat{a}/\hat{b})-\mu-2}.$$

By Lemma 3.7, $\hat{q}$ is $(\mu{+}1)$-exact, $expo(\hat{n_0}\hat{r_0}) \leq expo(\hat{q})$, and $|\hat{n_0}\hat{r_0} - \hat{q}| \leq 2^{expo(\hat{n_0}\hat{r_0})-\mu-1}$. We may now invoke Lemma 4.8 with $\zeta = \hat{n_0}\hat{r_0}$, which yields the desired inequality.

Thus, we may assume that $pc \neq$ `PC-32`. Now

$$\hat{d_0}\hat{r_0} < (\hat{b}\xi_0 + \alpha)(2 - \hat{b}\xi_0) = \hat{b}\xi_1 + \alpha(2 - \hat{b}\xi_0) < \hat{b}\xi_1 + \alpha + 2^{-14}\alpha,$$

$$\begin{aligned}\hat{d_0}\hat{r_0} &\geq (\hat{b}\xi_0 - \alpha)(2 - \hat{b}\xi_0 - 2\alpha) = \hat{b}\xi_1 - 2\alpha\hat{b}\xi_0 - \alpha(2 - \hat{b}\xi_0) + 2\alpha^2 \\ &> \hat{b}\xi_1 - 2\alpha(1+\epsilon) - \alpha(1+\epsilon) > \hat{b}\xi_1 - 3\alpha - 2^{-13}\alpha,\end{aligned}$$

and Lemma 2.26 implies

$$|\hat{d_1} - \hat{d_0}\hat{r_0}| \leq 2^{expo(\hat{d_0}\hat{r_0})-M} \leq \alpha,$$

hence

$$\hat{d_1} \leq \hat{d_0}\hat{r_0} + \alpha < \hat{b}\xi_1 + 2\alpha + 2^{-14}\alpha$$

and

$$\hat{d_1} \geq \hat{d_0}\hat{r_0} - \alpha > \hat{b}\xi_1 - 4\alpha - 2^{-13}\alpha.$$

By Lemmas 3.5 and 3.6,

$$\hat{r_1} < 2 - \hat{d_0}\hat{r_0} < (2 - \hat{b}\xi_1) + 3\alpha + 2^{-13}\alpha$$

and

$$\hat{r_1} \geq 2 - \hat{d_0}\hat{r_0} - 2\alpha > (2 - \hat{b}\xi_1) - 3\alpha - 2^{-14}\alpha > 0.$$

Continuing in this manner, we have

$$|\hat{n_1} - \hat{n_0}\hat{r_0}| \leq 2^{expo(\hat{n_0}\hat{r_0})-M} \leq 2\alpha\beta,$$

$$\hat{n_1} \leq \hat{n_0}\hat{r_0} + 2\alpha\beta < \hat{a}\xi_1 + 4\alpha\beta + 2^{-13}\alpha\beta,$$

$$\hat{n_1} \geq \hat{n_0}\hat{r_0} - 2\alpha\beta > \hat{a}\xi_1 - 8\alpha\beta - 2^{-12}\alpha\beta,$$

$$\begin{aligned}\hat{n_1}\hat{r_1} &< (\hat{a}\xi_1 + 4\alpha\beta + 2^{-13}\alpha\beta)((2 - \hat{b}\xi_1) + 3\alpha + 2^{-13}\alpha) \\ &< \hat{a}\xi_2 + (4\alpha\beta + 2^{-13}\alpha\beta)(1 + \epsilon^2) + 2\beta(1 + \epsilon^2)(3\alpha + 2^{-13}\alpha) \\ &\quad +(4\alpha\beta + 2^{-13}\alpha\beta)(3\alpha + 2^{-13}\alpha) \\ &< \hat{a}\xi_2 + 10\alpha\beta + 2^{-11}\alpha\beta,\end{aligned}$$

$$\begin{aligned}
\hat{n_1}\hat{r_1} &> (\hat{a}\xi_1 - 8\alpha\beta - 2^{-12}\alpha\beta)((2-\hat{b}\xi_1) - 3\alpha + 2^{-14}\alpha) \\
&> \hat{a}\xi_2 - (8\alpha\beta + 2^{-12}\alpha\beta)(1+\epsilon^2) - 2\beta(1+\epsilon^2)(3\alpha + 2^{-14}\alpha) \\
&> \hat{a}\xi_2 - 14\alpha\beta - 2^{-11}\alpha\beta,
\end{aligned}$$

and

$$\begin{aligned}
|\hat{n_1}\hat{r_1} - \hat{a}/\hat{b}| &\leq |\hat{n_1}\hat{r_1} - \hat{a}\xi_2| + |\hat{a}\xi_2 - \hat{a}/\hat{b}| < 15\alpha\beta + 2^{-11}\alpha\beta + 2\beta\epsilon^4 \\
&< (15 \cdot 2^{-75} + 81 \cdot 2^{-63})\beta < 2^{-56}\beta \\
&= 2^{expo(\hat{a}/\hat{b})-56}.
\end{aligned}$$

Suppose $pc = \texttt{PC-64}$, and therefore $\mu = 53$. Then

$$|\hat{n_1}\hat{r_1} - \hat{a}/\hat{b}| < 2^{expo(\hat{a}/\hat{b})-56} < 2^{expo(\hat{a}/\hat{b})-\mu-2}.$$

The remaining hypotheses of Lemma 4.8, with $\hat{n_1}\hat{r_1}$ substituted for ζ, again follow from Lemma 3.7, and the desired inequality follows.

Thus, we may assume $pc = \texttt{PC-80}$ or $pc = \texttt{PC-87}$. Continuing, we have

$$\begin{aligned}
\hat{d}_1\hat{r_1} &< (\hat{b}\xi_1 + 2\alpha + 2^{-14}\alpha)((2-\hat{b}\xi_1) + 3\alpha + 2^{-13}\alpha) \\
&< \hat{b}\xi_2 + (2\alpha + 2^{-14}\alpha)(1+\epsilon^2) + (3\alpha + 2^{-13}\alpha)(1+\epsilon^2) \\
&\quad +(2\alpha + 2^{-14}\alpha)(3\alpha + 2^{-13}\alpha) \\
&< \hat{b}\xi_2 + 5\alpha + 2^{-12}\alpha,
\end{aligned}$$

$$\begin{aligned}
\hat{d}_1\hat{r_1} &> (\hat{b}\xi_1 - 4\alpha - 2^{-13}\alpha)((2-\hat{b}\xi_1) - 3\alpha - 2^{-14}\alpha) \\
&> \hat{b}\xi_2 - (4\alpha + 2^{-13}\alpha)(1+\epsilon^2) - (1+\epsilon^2)(3\alpha + 2^{-14}\alpha) \\
&> \hat{b}\xi_2 - 7\alpha - 2^{-12}\alpha,
\end{aligned}$$

$$\hat{r_2} < 2 - \hat{d}_1\hat{r_1} < (2 - \hat{b}\xi_2) + 7\alpha + 2^{-12}\alpha,$$

$$\hat{r_2} \geq 2 - \hat{d}_1\hat{r_1} - 2\alpha > (2 - \hat{b}\xi_2) - 7\alpha - 2^{-12}\alpha > 0,$$

$$|\hat{n_2} - \hat{n_1}\hat{r_1}| \leq 2^{expo(\hat{n_1}\hat{r_1})-M} \leq 2\alpha\beta,$$

$$\hat{n_2} \leq \hat{n_1}\hat{r_1} + 2\alpha\beta < \hat{a}\xi_2 + 12\alpha\beta + 2^{-11}\alpha\beta,$$

$$\hat{n_2} \geq \hat{n_1}\hat{r_1} - 2\alpha\beta > \hat{a}\xi_2 - 16\alpha\beta - 2^{-11}\alpha\beta,$$

$$\begin{aligned}
\hat{n_2}\hat{r_2} &< (\hat{a}\xi_2 + 12\alpha\beta + 2^{-11}\alpha\beta)((2-\hat{b}\xi_2) + 7\alpha + 2^{-12}\alpha) \\
&< \hat{a}\xi_3 + (12\alpha\beta + 2^{-11}\alpha\beta)(1+\epsilon^4) + 2\beta(1+\epsilon^4)(7\alpha + 2^{-12}\alpha) \\
&\quad +(12\alpha\beta + 2^{-11}\alpha\beta)(7\alpha + 2^{-12}\alpha) \\
&< \hat{a}\xi_3 + 26\alpha\beta + 2^{-9}\alpha\beta,
\end{aligned}$$

and

$$\begin{aligned}
\hat{n_2}\hat{r_2} &> (\hat{a}\xi_2 - 16\alpha\beta - 2^{-11}\alpha\beta)((2-\hat{b}\xi_2) - 7\alpha + 2^{-12}\alpha) \\
&> \hat{a}\xi_3 - (16\alpha\beta + 2^{-11}\alpha\beta)(1+\epsilon^4) - 2\beta(1+\epsilon^4)(7\alpha 2 + 2^{-12}\alpha) \\
&> \hat{a}\xi_3 - 30\alpha\beta - 2^{-9}\alpha\beta.
\end{aligned}$$

Finally, since $\mu \leq 68$,

$$\begin{aligned}
|\hat{n_2}\hat{r_2} - \hat{a}/\hat{b}| &\leq |\hat{n_2}\hat{r_2} - \hat{a}\xi_3| + |\hat{a}\xi_3 - \hat{a}/\hat{b}| < 31\alpha\beta + 2\beta\epsilon^8 \\
&< (30 \cdot 2^{-75} + 81 \cdot 2^{-110})\beta < 2^{-70}\beta \\
&\leq 2^{expo(\hat{a}/\hat{b})-\mu-2},
\end{aligned}$$

and the lemma follows from Lemma 4.8, with $\zeta = \hat{n_2}\hat{r_2}$. □

4.4 The Operation OP-SQRT

The Newton-Raphson formula for approximating $1/\sqrt{\hat{b}}$ is

$$\xi_i = \frac{\xi_{i-1}}{2}(3 - \hat{b}\xi_{i-1}^2).$$

Since the relative error of this approximation is

$$\left|\frac{\xi_i - 1/\sqrt{\hat{b}}}{1/\sqrt{\hat{b}}}\right| = |\sqrt{\hat{b}}\xi_i - 1| < |\sqrt{\hat{b}}\xi_i - 1||\sqrt{\hat{b}}\xi_i + 1| = |\hat{b}\xi^2 - 1|,$$

convergence is established by the following lemma, which is proved in [8]:

Lemma 4.10 *Let $b, x \in \mathbb{Q}$ with $0 \leq bx^2 \leq 4$ and let $y = \frac{x}{2}(3 - bx^2)$. Then*

$$0 \leq 1 - by^2 \leq (1 - bx^2)^2.$$

We shall use Lemma 4.10 to derive an error estimate for q in the OP-SQRT case.

Lemma 4.11 *For all $i \in \mathbb{N}$, let ξ_i be defined by*

$$\xi_i = \begin{cases} \hat{x_0} & \textit{if } i = 0 \\ \frac{\xi_{i-1}}{2}(3 - \hat{b}\xi_{i-1}^2) & \textit{if } i > 0, \end{cases}$$

and let $\epsilon = 3/2^{16}$. Assume that $\hat{q} > 0$ and $\hat{q}$ is $(\mu + 1)$-exact, where $\mu \geq 24$.

Let $\ell, h \in \mathbb{Q}$ such that $0 \leq \ell \leq h$ and $\ell^2 \leq \hat{b} \leq h^2$. Let $\zeta, \eta \in \mathbb{Q}^+$ and $i \in \mathbb{Z}^+$ such that

$$expo(\zeta) \leq expo(\hat{q}),$$

$$|\hat{q} - \zeta| \leq 2^{expo(\zeta)-\mu-1},$$

$$|\hat{b}\xi_i - \zeta| < 2^{\lfloor expo(\hat{b})/2 \rfloor}\eta,$$

and

$$2\eta + 8\epsilon^{2^i} \leq 2^{-\mu-1}.$$

Then

$$h > q - 2^{min(expo(\hat{q}),expo(h))-\mu}$$

and

$$\ell < q + 2^{min(expo(\hat{q}),expo(\ell))-\mu}.$$

Proof: By Lemmas 4.6 and 4.10, $0 \le 1 - \hat{b}\xi_i^2 < \epsilon^{2^i}$, where $\epsilon = 3/2^{16}$, and hence

$$(\hat{b}\xi_i)^2 = \hat{b}(\hat{b}\xi_i^2) > \hat{b}(1-\epsilon^{2^i}) > 2^{expo(\hat{b})-1} > (2^{\lfloor expo(\hat{b})/2\rfloor-1})^2$$

and $\hat{b}\xi_i > 2^{\lfloor expo(\hat{b})/2\rfloor-1}$.

Since $|\hat{q}-\zeta| \le 2^{expo(\zeta)-\mu-1} \le \zeta/4$, $\hat{q} \ge \frac{3}{4}\zeta$. Since $\eta < 2^{-\mu-2}$,

$$|\hat{b}\xi_i - \zeta| < 2^{\lfloor expo(\hat{b})/2\rfloor-\mu-2} < \hat{b}\xi_i 2^{-\mu-1} \le \hat{b}\xi_i/4,$$

and hence $\hat{q} \ge \frac{3}{4}\zeta > \frac{9}{16}\hat{b}\xi_i$, which implies

$$\hat{q}^2 > \frac{81}{256}(\hat{b}\xi_i)^2 > \frac{81}{256}\hat{b}(1-\epsilon^{2^i}) > \hat{b}/4.$$

It follows that $expo(\hat{q}) \ge \lfloor expo(\hat{b})/2\rfloor - 1$.

Since $h^2 \ge \hat{b} \ge \hat{b}(\hat{b}\xi_i^2) = (\hat{b}\xi_i)^2$,

$$h \ge \hat{b}\xi_i \ge \hat{q} - (|\hat{q}-\zeta| + |\hat{b}\xi_i - \zeta|) > \hat{q} - (2^{expo(\zeta)-\mu-1} + 2^{expo(\hat{q})-\mu-1}) \ge \hat{q} - 2^{expo(\hat{q})-\mu}.$$

Therefore, we may assume $expo(h) < expo(\hat{q})$. But $|h| > |\hat{q}|/2$, hence $expo(h) = expo(\hat{q}) - 1$. Also note that $expo(h) \ge \lfloor expo(\hat{b})/2\rfloor$, for otherwise $h < 2^{\lfloor expo(\hat{b})/2\rfloor}$ and

$$\hat{b} \le h^2 < 2^{2\lfloor expo(\hat{b})/2\rfloor} \le 2^{expo(\hat{b})}.$$

We may further assume $expo(\zeta) = expo(\hat{q})$, for otherwise $expo(\zeta) \le expo(h)$ and

$$\begin{aligned} h &\ge \hat{b}\xi_i \ge \hat{q} - (|\hat{q}-\zeta| + |\hat{b}\xi_i - \zeta|) \\ &> \hat{q} - (2^{expo(\zeta)-\mu-1} + 2^{\lfloor expo(\hat{b})/2\rfloor-\mu-2}) \ge \hat{q} - 2^{expo(h)-\mu}. \end{aligned}$$

If $\hat{q} > 2^{expo(\hat{q})}$, then $\hat{q} \ge 2^{expo(\hat{q})} + 2^{expo(\hat{q})-\mu}$ by Lemma 2.13, and

$$h > \hat{q} - 2^{expo(\hat{q})-\mu} \ge 2^{expo(\hat{q})} = 2^{expo(h)+1}.$$

Therefore, $\hat{q} = 2^{expo(\hat{q})}$, which implies $\zeta \ge \hat{q}$ and

$$h \ge \hat{b}\xi_i = \hat{q} - (\hat{q} - \hat{b}\xi_i) \ge \hat{q} - (\zeta - \hat{b}\xi_i) > \hat{q} - 2^{\lfloor expo(\hat{b})/2\rfloor-\mu-2} \ge \hat{q} - 2^{expo(h)-\mu}.$$

In order to derive the bound for ℓ, we may assume $expo(\hat{q}) \le expo(\ell)$, for otherwise $\ell < \hat{q}$ and the inequality holds trivially. Since $(\hat{b}\xi_i)^2 > \hat{b}(1-\epsilon^{2^i})$,

$$\ell^2 \le \hat{b} < (\hat{b}\xi_i)^2/(1-\epsilon^{2^i}) < [\hat{b}\xi_i/(1-\epsilon^{2^i})]^2,$$

and hence

$$\ell < \hat{b}\xi_i/(1-\epsilon^{2^i}) < \hat{b}\xi_i(1+2\epsilon^{2^i}).$$

Recall that $expo(\hat{q}) \ge \lfloor expo(\hat{b})/2\rfloor - 1$ and $\hat{q} > \frac{9}{16}\hat{b}\xi_i$, hence $\hat{b}\xi_i < 2^{expo(\hat{q})+2}$. Thus,

$$\begin{aligned} \ell &< \hat{b}\xi_i(1+2\epsilon^{2^i}) < \hat{b}\xi_i + 8\epsilon^{2^i}2^{expo(\hat{q})} \le \hat{q} + |\hat{q}-\zeta| + |\zeta - \hat{b}\xi_i| + 8\epsilon^{2^i}2^{expo(\hat{q})} \\ &< \hat{q} + 2^{expo(\hat{q})-\mu-1} + 2^{\lfloor expo(\hat{b})/2\rfloor}\eta + 8\epsilon^{2^i}2^{expo(\hat{q})} \le \hat{q} + 2^{expo(\hat{q})}(2^{-\mu-1} + 2\eta + 8\epsilon^{2^i}) \\ &\le \hat{q} + 2^{expo(\hat{q})}(2^{-\mu-1} + 2^{-\mu-1}) = \hat{q} + 2^{expo(\hat{q})-\mu}.\square \end{aligned}$$

We shall also require the following lemma, in order to invoke Lemma 3.8.

Lemma 4.12 *Under the hypothesis of Lemma 4.11,* $|\hat{q}^2 - \hat{b}| < 2^{expo(\hat{b})-3}$.

Proof: Since $expo(\hat{b}) \leq 2\lfloor expo(\hat{b})/2 \rfloor + 1$, $\hat{b} < 2^{expo(\hat{b})+1} \leq (2^{\lfloor expo(\hat{b})/2 \rfloor + 1})^2$. Thus,

$$(\hat{b}\xi_i)^2 = \hat{b}(\hat{b}\xi_i^2) \leq \hat{b} < (2^{\lfloor expo(\hat{b})/2 \rfloor + 1})^2$$

and $\hat{b}\xi_i < 2^{\lfloor expo(\hat{b})/2 \rfloor + 1}$. Now since

$$|\hat{q} - \hat{b}\xi_i| \leq |\hat{q} - \zeta| + |\hat{b}\xi_i - \zeta| < 2^{expo(\hat{q})-\mu} \leq 2^{\lfloor expo(\hat{b})/2 \rfloor + 1 - \mu}$$

and

$$|\hat{q} + \hat{b}\xi_i| \leq 2\hat{b}\xi_i + |\hat{q} - \hat{b}\xi_i| < 2^{\lfloor expo(\hat{b})/2 \rfloor + 2} + 2^{\lfloor expo(\hat{b})/2 \rfloor + 1 - \mu} < 2^{\lfloor expo(\hat{b})/2 \rfloor + 3},$$

we have

$$|\hat{q}^2 - (\hat{b}\xi_i)^2| = |\hat{q} - \hat{b}\xi_i||\hat{q} + \hat{b}\xi_i| < 2^{2\lfloor expo(\hat{b})/2 \rfloor + 4 - \mu} \leq \hat{b}2^{4-\mu}.$$

Thus,

$$|\hat{q}^2 - \hat{b}| \leq |\hat{q}^2 - (\hat{b}\xi_i)^2| + \hat{b}|1 - \hat{b}\xi_i^2| < \hat{b}2^{5-\mu} < 2^{expo(\hat{b})+6-\mu}.\square$$

Lemma 4.13 *Assume* $op =$ `OP-SQRT`, $\hat{b} > 0$, $expo(\hat{b}) \leq 2^{17} - 2$, *and let* $mbits(pc) = \mu$. *Let* $\ell, h \in \mathbb{Q}$ *such that* $0 \leq \ell \leq h$ *and* $\ell^2 \leq \hat{b} \leq h^2$. *Then* q *is normal,* $\hat{q}$ *is* $(\mu + 1)$*-exact,*

$$\ell < \hat{q} + 2^{min(expo(\hat{q}),expo(\ell))-\mu},$$

$$h > \hat{q} - 2^{min(expo(\hat{q}),expo(h))-\mu},$$

and

$$|\hat{q}^2 - \hat{b}| < 2^{expo(\hat{b})-3}.$$

Proof: Let $\alpha = 2^{-M}$, $\beta = 2^{\lfloor expo(\hat{b})/2 \rfloor}$, and $\epsilon = 3/2^{16}$. For $i \in \mathbb{N}$, let ξ_i be defined as in Lemma 4.11. Then $\hat{b} < 4\beta^2$ and $|1 - \hat{b}\xi_i^2| < \epsilon^{2^i}$. For $i > 0$, $\hat{b}\xi_i^2 \leq 1$ and $\hat{b}\xi_i < 2\beta$, which implies $2^{expo(\hat{b}\xi_i)} \leq \beta$. On the other hand,

$$(\hat{b}\xi_0)^2 = \hat{b}(\hat{b}\xi_0^2) < 4\beta^2(1 + \epsilon) < (2\beta(1 + \epsilon))^2,$$

hence $\hat{b}\xi_0 < 2\beta(1 + \epsilon) < 4\beta$, which implies $2^{expo(\hat{b}\xi_0)} \leq 2\beta$. Also note that for all i,

$$(3 - \hat{b}\xi_i^2)/2 = 1 + (1 - \hat{b}\xi_i^2)/2 < 1 + \epsilon^{2^i}/2$$

.

We proceed as in the proof of Lemma 4.9, invoking Lemmas 4.11 and 4.12 in each of several cases. According to Lemma 4.6(c), the hypothesis of Theorem 1 is satisfied by $x = y = \hat{x_0}$. Thus,

$$\hat{t_0} = near(\hat{x_0}^2, M) = \hat{x_0}^2 = \xi_0^2.$$

Similarly,

$$\hat{d_0} = near(\hat{b}\xi_0, M)$$

and

$$\hat{n_0} = near(\hat{b}\hat{t_0}, M) = near(\hat{b}\xi_0^2, M).$$

Therefore, by Lemma 2.26,

$$|\hat{d}_0 - \hat{b}\xi_0| \leq 2^{expo(\hat{b}\xi_0)-M} \leq 2\alpha\beta$$

and

$$|\hat{n_0} - \hat{b}\xi_0^2| \leq 2^{expo(\hat{b}\xi_0^2)-M} \leq \alpha.$$

By Lemmas 3.5 and 4.6,

$$(3 - \hat{b}\xi_0^2)/2 - 2\alpha \leq \hat{r_0} < (3 - \hat{b}\xi_0^2)/2.$$

Thus,

$$\hat{d}_0\hat{r_0} < (\hat{b}\xi_0 + 2\alpha\beta)(3 - \hat{b}\xi_0^2)/2 < \hat{b}\xi_1 + 2\alpha\beta(1 + \epsilon/2) < \hat{b}\xi_1 + 2\alpha\beta + 2^{-14}\alpha\beta$$

and

$$\begin{array}{rcl} \hat{d}_0\hat{r_0} & > & (\hat{b}\xi_0 - 2\alpha\beta)((3 - \hat{b}\xi_0^2)/2 - 2\alpha) > \hat{b}\xi_1 - 2\alpha\beta(1 + \epsilon/2) - 2\beta(1 + \epsilon)2\alpha \\ & > & \hat{b}\xi_1 - 6\alpha\beta - 2^{-12}\alpha\beta. \end{array}$$

Suppose $pc = \mathtt{PC\text{-}32}$ and $\mu = 24$. We shall apply Lemmas 4.11 and 4.12 with $\zeta = \hat{d}_0\hat{r_0}$, $i = 1$, and $\eta = 7\alpha$. Under these substitutions, we have

$$|\hat{b}\xi_i - \zeta| < 7\alpha\beta = 2^{\lfloor expo(\hat{b})/2 \rfloor}\eta$$

and

$$2\eta + 8\epsilon^{2^i} = 14 \cdot 2^{-M} + 8\epsilon^2 \leq 14 \cdot 2^{-75} + 9 \cdot 2^{-29} < 2^{-25} = 2^{-\mu-1}.$$

The remaining hypotheses of Lemma 4.11 are ensured by Lemma 3.7, and the conclusion follows.

Thus, we may assume $pc \neq \mathtt{PC\text{-}32}$. Now we have $\hat{t_1} = near(\hat{r_0}^2, M)$, hence $|\hat{t_1} - \hat{r_0}^2| \leq \alpha$, which implies

$$\hat{t_1} \leq (3 - \hat{b}\xi_0^2)^2/4 + \alpha$$

and

$$\begin{array}{rcl} \hat{t_1} & \geq & ((3 - \hat{b}\xi_0^2)/2 - 2\alpha)^2 - \alpha \\ & > & (3 - \hat{b}\xi_0^2)^2/4 - 4\alpha(1 + \epsilon/2) - \alpha \\ & > & (3 - \hat{b}\xi_0^2)^2/4 - 5\alpha - 2^{-13}\alpha. \end{array}$$

Consequently,

$$\begin{array}{rcl} \hat{n_0}\hat{t_1} & \leq & (b\xi_0^2 + \alpha)((3 - \hat{b}\xi_0^2)^2/4 + \alpha) < \hat{b}\xi_1^2 + (1 + \epsilon)\alpha + \alpha(1 + \epsilon/2)^2 + \alpha^2 \\ & < & \hat{b}\xi_1^2 + 2\alpha + 2^{-13}\alpha \end{array}$$

and

$$\begin{array}{rcl} \hat{n_0}\hat{t_1} & \geq & (b\xi_0^2 - \alpha)((3 - \hat{b}\xi_0^2)^2/4 - 5\alpha - 2^{-13}\alpha) \\ & > & \hat{b}\xi_1^2 - (1 + \epsilon)(5\alpha + 2^{-13}\alpha) - \alpha(1 + \epsilon/2)^2 > \hat{b}\xi_1^2 - 6\alpha - 2^{-11}\alpha. \end{array}$$

Since $\hat{d}_1 = near(\hat{d}_0\hat{r_0}, M)$, $|\hat{d}_1 - \hat{d}_0\hat{r_0}| \leq 2^{expo(\hat{d}_0\hat{r_0})-M} \leq 2\alpha\beta$, hence

$$\hat{b}\xi_1 - 8\alpha\beta - 2^{-12}\alpha\beta < \hat{d}_1 < \hat{b}\xi_1 + 4\alpha\beta + 2^{-14}\alpha\beta.$$

Similarly, $\hat{n_1} = near(\hat{n_0}\hat{t_1}, M)$, $|\hat{n_1} - \hat{n_0}\hat{t_1}| \leq 2^{expo(\hat{n_0}\hat{t_1})-M} \leq \alpha$, and

$$\hat{b}\xi_1^2 - 7\alpha - 2^{-11}\alpha < \hat{n_1} < \hat{b}\xi_1^2 + 3\alpha + 2^{-13}\alpha.$$

By Lemmas 3.5 and 3.6,

$$\hat{r_1} < (3 - \hat{n_0}\hat{t_1})/2 < (3 - \hat{b}\xi_1^2)/2 + 3\alpha + 2^{-10}\alpha$$

and

$$\hat{r_1} \geq (3 - \hat{n_0}\hat{t_1})/2 - 2\alpha > (3 - \hat{b}\xi_1^2)/2 - 3\alpha - 2^{-12}\alpha.$$

Thus,

$$\begin{array}{rcl} \hat{d_1}\hat{r_1} & < & (\hat{b}\xi_1 + 4\alpha\beta + 2^{-14}\alpha\beta)((3 - \hat{b}\xi_1^2)/2 + 3\alpha + 2^{-10}\alpha) \\ & < & \hat{b}\xi_2 + 2\beta(3\alpha + 2^{-10}\alpha) + (4\alpha\beta + 2^{-14}\alpha\beta)(1 + \epsilon^2/2) \\ & & +(3\alpha + 2^{-10}\alpha)(4\alpha\beta + 2^{-14}\alpha\beta) \\ & < & \hat{b}\xi_2 + 10\alpha\beta + 2^{-8}\alpha\beta \end{array}$$

and

$$\begin{array}{rcl} \hat{d_1}\hat{r_1} & > & (\hat{b}\xi_1 - 8\alpha\beta - 2^{-12}\alpha\beta)((3 - \hat{b}\xi_1^2)/2 - 3\alpha - 2^{-12}\alpha) \\ & > & \hat{b}\xi_2 - 2\beta(3\alpha + 2^{-12}\alpha) - (8\alpha\beta + 2^{-12}\alpha\beta)(1 + \epsilon^2/2) \\ & > & \hat{b}\xi_2 - 14\alpha\beta - 2^{-10}\alpha\beta. \end{array}$$

Suppose pc = `PC-64` and $\mu = 53$. We shall again invoke Lemmas 4.11 and 4.12, now with $\zeta = \hat{d_1}\hat{r_1}$, $i = 2$, and $\eta = 15\alpha$. Thus

$$|\hat{b}\xi_i - \zeta| < 15\alpha\beta = 2^{\lfloor expo(\hat{b})/2 \rfloor}\eta$$

and

$$2\eta + 8\epsilon^{2^i} = 30 \cdot 2^{-M} + 8\epsilon^4 \leq 30 \cdot 2^{-75} + 81 \cdot 2^{-61} < 2^{-54} = 2^{-\mu-1}.$$

The remaining hyptheses of Lemma 4.11 are again ensured by Lemma 3.7.

Thus, we may assume pc = `PC-80` or pc = `PC-87`. Continuing in the same manner, we have

$$|\hat{t_2} - \hat{r_1}^2| \leq 2^{expo(\hat{r_1}^2)-M} \leq \alpha,$$

$$\begin{array}{rcl} \hat{t_2} & < & (3 - \hat{b}\xi_1^2)^2/4 + 2(1 + \epsilon^2/2)^2(3\alpha + 2^{-10}\alpha) + (3\alpha + 2^{-10}\alpha)^2 + \alpha \\ & < & (3 - \hat{b}\xi_1^2)^2/4 + 7\alpha + 2^{-8}\alpha, \end{array}$$

$$\begin{array}{rcl} \hat{t_2} & > & (3 - \hat{b}\xi_1^2)^2/4 - 2(1 + \epsilon^2/2)^2(3\alpha + 2^{-12}\alpha) - \alpha \\ & > & (3 - \hat{b}\xi_1^2)^2/4 - 7\alpha + 2^{-10}\alpha, \end{array}$$

$$|\hat{d_2} - \hat{d_1}\hat{r_1}| \leq 2^{expo(\hat{d_1}\hat{r_1})-M} \leq 2\alpha\beta,$$

$$\hat{b}\xi_2 - 16\alpha\beta - 2^{-10}\alpha\beta < \hat{d_2} < \hat{b}\xi_2 + 12\alpha\beta + 2^{-8}\alpha\beta,$$

$$\begin{aligned}
\hat{n_1}\hat{t_2} &< (\hat{b}\xi_1^2 + 3\alpha + 2^{-13}\alpha)((3 - \hat{b}\xi_1^2)^2/4 + 7\alpha + 2^{-8}\alpha) \\
&< \hat{b}\xi_2^2 + (7\alpha + 2^{-8}\alpha) + (1 + \epsilon^2/2)^2(3\alpha + 2^{-13}\alpha) \\
&\quad +(3\alpha + 2^{-13}\alpha)(7\alpha + 2^{-8}\alpha) \\
&< \hat{b}\xi_1^2 + 10\alpha + 2^{-7}\alpha,
\end{aligned}$$

$$\begin{aligned}
\hat{n_1}\hat{t_2} &> (\hat{b}\xi_1^2 - 7\alpha - 2^{-11}\alpha)((3 - \hat{b}\xi_1^2)^2/4 - 7\alpha + 2^{-10}\alpha) \\
&> \hat{b}\xi_2^2 - (7\alpha + 2^{-10}\alpha) - (1 + \epsilon^2/2)^2(7\alpha + 2^{-11}\alpha) \\
&> \hat{b}\xi_1^2 - 14\alpha - 2^{-9}\alpha,
\end{aligned}$$

$$\hat{r_2} < (3 - \hat{n_1}\hat{t_2})/2 < (3 - \hat{b}\xi_2^2)/2 + 7\alpha + 2^{-10}\alpha,$$

$$\hat{r_2} \geq (3 - \hat{n_1}\hat{t_2})/2 - 2\alpha > (3 - \hat{b}\xi_2^2)/2 - 7\alpha - 2^{-8}\alpha,$$

$$\begin{aligned}
\hat{d_2}\hat{r_2} &< (\hat{b}\xi_2 + 12\alpha\beta + 2^{-8}\alpha\beta)((3 - \hat{b}\xi_2^2)/2 + 7\alpha + 2^{-10}\alpha) \\
&< \hat{b}\xi_3 + 2\beta(7\alpha + 2^{-10}\alpha) + (1 + \epsilon^4/2)(12\alpha\beta + 2^{-8}\alpha\beta) \\
&\quad +(12\alpha\beta + 2^{-8}\alpha\beta)(7\alpha + 2^{-10}\alpha) \\
&< \hat{b}\xi_3 + 26\alpha\beta + 2^{-7}\alpha\beta,
\end{aligned}$$

and

$$\begin{aligned}
\hat{d_2}\hat{r_2} &> (\hat{b}\xi_2 - 16\alpha\beta - 2^{-10}\alpha\beta)((3 - \hat{b}\xi_2^2)/2 - 7\alpha + 2^{-8}\alpha) \\
&> \hat{b}\xi_3 - 2\beta(7\alpha + 2^{-8}\alpha) - (1 + \epsilon^4/2)(16\alpha\beta + 2^{-10}\alpha\beta) \\
&> \hat{b}\xi_3 - 30\alpha\beta - 2^{-6}\alpha\beta.
\end{aligned}$$

Finally, we apply Lemmas 4.11 and 4.12 with $\zeta = \hat{d_2}\hat{r_2}$, $i = 3$, and $\eta = 31\alpha$. Thus,

$$|\hat{b}\xi_i - \zeta| < 31\alpha\beta = 2^{\lfloor expo(\hat{b})/2 \rfloor}\eta,$$

and since $\mu \leq 68$,

$$2\eta + 8\epsilon^{2^i} = 62 \cdot 2^{-M} + 8\epsilon^8 \leq 62 \cdot 2^{-75} + 2^{-112} < 2^{-69} \leq 2^{-\mu-1}.$$

The proof is completed by invoking Lemmas 3.7 and 4.11. □

4.5 Final Rounding

The remaining analysis pertains to the latter part of *FPU-DIV-SQRT*, in which the approximation q is adjusted to produce the correctly rounded result.

The significance of the variables *q-guard* and *q-lsb* is given by the following:

Lemma 4.14 *Assume that q is normal and $\hat{q}$ is $(\mu + 1)$-exact, where $\mu = mbits(pc)$.*

(a) q-guard $= 0 \Leftrightarrow \hat{q}$ is μ-exact;

(b) q-lsb $= 0 \Leftrightarrow trunc(\hat{q}, \mu)$ is $(\mu - 1)$-exact.

Proof: (a) Let $m = \mathit{get\text{-}man}(q)$. Then m is $(\mu+1)$-exact, i.e,

$$m2^{\mu-expo(m)} = m2^{\mu+1-M} \in \mathbb{Z}$$

and

$$\mathit{q\text{-}guard} = m[M-\mu-1] = rem(\lfloor m2^{\mu+1-M} \rfloor, 2) = rem(m2^{\mu+1-M}, 2).$$

But

$$m \text{ is } \mu\text{-exact} \Leftrightarrow m2^{\mu-M} \in \mathbb{Z} \Leftrightarrow m2^{\mu+1-M} \text{ is even} \Leftrightarrow \mathit{q\text{-}guard} = 0.$$

(b) $\mathit{q\text{-}lsb} = m[M-\mu] = rem(\lfloor m2^{\mu-M} \rfloor, 2)$ and $trunc(m,\mu) = \lfloor m2^{\mu-M} \rfloor 2^{M-\mu}$. Thus,

$$\begin{aligned} trunc(m,\mu) \text{ is } (\mu-1)\text{-exact} &\Leftrightarrow \lfloor m2^{\mu-M} \rfloor 2^{M-\mu} 2^{(\mu-1)-1-(M-1)} = \lfloor m2^{\mu-M} \rfloor / 2 \in \mathbb{Z} \\ &\Leftrightarrow \lfloor m2^{\mu-M} \rfloor \text{ is even} \\ &\Leftrightarrow \mathit{q\text{-}lsb} = 0. \Box \end{aligned}$$

The correctness proof for division will be based on the following:

Lemma 4.15 *Let $\mu = mbits(pc)$. Suppose q is normal, $\hat{q}$ is $(\mu+1)$-exact, $sign = 0$, and $2^{1-2^{17}} < \hat{q} < 2^{2^{17}}(2-2^{1-\mu})$. Let $x \in \mathbb{Q}$ such that*

(a) $|x - \hat{q}| < 2^{min(expo(\hat{q}),expo(x))-\mu}$;

(b) if $\mathit{rem\text{-}neg} = 1$, then $\hat{q} > x$;

(c) if $\mathit{rem\text{-}pos} = 1$, then $\hat{q} < x$;

(d) if $\mathit{rem\text{-}zero} = 1$, then $\hat{q} = x$.

Then z is normal and $rnd(x, rc, pc) = \hat{z}$.

Proof: Note that the hypothesis implies that $\hat{z} > 0$ and $x > 0$.

Case 1: $rc =$ `RC-NEG` or $rc =$ `RC-CHOP`

In this case, $rnd(x, rc, pc) = trunc(x, \mu)$.

Subcase 1.1: $\mathit{q\text{-}guard} = 0$ and $\mathit{rem\text{-}neg} = 1$

By Lemma 4.14, $\hat{q}$ is μ-exact. Also, $x < \hat{q}$. By Lemma 2.24,

$$\mathit{get\text{-}man}(q) \;\mathtt{\&}\; (2^M - 2^{M-\mu}) = trunc(\mathit{get\text{-}man}(q), \mu) = \mathit{get\text{-}man}(q).$$

If $\mathit{get\text{-}man}(q) = 2^{M-1}$, then $\hat{q} = 2^{expo(\hat{q})}$, where by hypothesis, $expo(\hat{q}) > 1 - 2^{17}$. In this case, $\hat{z} = (2-2^{1-\mu})2^{expo(\hat{q})-1}$ and $expo(\hat{z}) = expo(\hat{q}) - 1$. In all other cases, $\hat{q} \geq 2^{expo(\hat{q})} + 2^{1+expo(\hat{q})-\mu}$, $\hat{z} = \hat{q} - 2^{1+expo(\hat{q})-\mu}$, $\hat{z} \geq 2^{expo(\hat{q})}$, and $expo(\hat{z}) = expo(\hat{q})$. In any case, $\hat{z} + 2^{1+expo(\hat{z})-\mu} = \hat{q}$. Since $trunc(x,\mu) \leq x < \hat{q}$, $trunc(x,\mu) \leq \hat{z}$ by Lemma 2.13. Also, $trunc(x,\mu) \geq \hat{z}$, for otherwise we would have $x < \hat{z}$, $expo(x) \leq expo(\hat{z})$, and

$$x > \hat{q} - 2^{expo(x)-\mu} > \hat{q} - 2^{1+expo(\hat{z})-\mu} = \hat{z}.$$

Subcase 1.2: $\mathit{q\text{-}guard} = 1$

In this case, $\hat{q}$ is not μ-exact, and $\hat{z} = trunc(\hat{q},\mu)$. By Lemma 2.27, $\hat{z} = \hat{q} - 2^{expo(\hat{q})-\mu}$. Therefore,

$$trunc(x,\mu) \leq x < \hat{q} + 2^{expo(\hat{q})-\mu} = \hat{z} + 2^{expo(\hat{q})+1-\mu} = \hat{z} + 2^{expo(\hat{z})+1-\mu},$$

and hence $trunc(x,\mu) \leq \hat{z}$. But since $x > \hat{q} - 2^{expo(\hat{q})-\mu} = \hat{z}$, $trunc(x,\mu) \geq trunc(\hat{z},\mu) = \hat{z}$.

Subcase 1.3: $q\text{-}guard = rem\text{-}neg = 0$

$\hat{q}$ is μ-exact, $x \geq \hat{q}$, and $\hat{z} = trunc(\hat{q}, \mu) = \hat{q}$.

In this case,

$$trunc(x, \mu) \leq x < \hat{q} + 2^{expo(\hat{q})-\mu} = \hat{z} + 2^{expo(\hat{z})-\mu} < \hat{z} + 2^{expo(\hat{z})+1-\mu},$$

which implies $trunc(x, \mu) \leq \hat{z}$. But $x \geq \hat{q} = \hat{z}$ implies $trunc(x, \mu) \geq \hat{z}$.

Case 2: $rc = \texttt{RC-POS}$

In this case, $rnd(x, rc, pc) = away(x, \mu)$.

Subcase 2.1: $q\text{-}guard = 1$

Here, $\hat{q}$ is $(\mu{+}1)$-exact but not μ-exact. By the same reasoning as used in Subcase 1.1, we may show that

$$\hat{z} = trunc(\hat{q}, \mu) + 2^{expo(\hat{q})+1-\mu}.$$

But then by Lemma 2.27,

$$\hat{z} = \hat{q} - 2^{expo(\hat{q})-\mu} + 2^{expo(\hat{q})+1-\mu} = \hat{q} + 2^{expo(\hat{q})-\mu} = away(\hat{q}, \mu).$$

Since $x < \hat{q} + 2^{expo(\hat{q})-\mu} = \hat{z}$, $away(x, \mu) \leq away(\hat{z}, \mu) = \hat{z}$. But $x > \hat{q} - 2^{expo(\hat{q})-\mu} = trunc(\hat{q}, \mu)$, hence $away(x, \mu) \geq trunc(\hat{q}, \mu) + 2^{expo(\hat{q})+1-\mu} = \hat{z}$.

Subcase 2.2: $q\text{-}guard = 0$ and $rem\text{-}pos = 1$.

In this case, $\hat{q}$ is μ-exact, $\hat{q} < x$, and

$$\hat{z} = trunc(\hat{q}, \mu) + 2^{expo(\hat{q})+1-\mu} = \hat{q} + 2^{expo(\hat{q})+1-\mu}.$$

Since $x < \hat{z}$, $away(x, \mu) \leq away(\hat{z}, \mu) = \hat{z}$. But $away(x, \mu) \geq x > \hat{q}$, so $away(x, \mu) \geq \hat{q} + 2^{expo(\hat{q})+1-\mu} = \hat{z}$.

Subcase 2.3: $q\text{-}guard = rem\text{-}pos = 0$

$\hat{q}$ is μ-exact, $x \leq \hat{q}$, and $\hat{z} = trunc(\hat{q}, \mu) = \hat{q}$. Thus,

$$away(x, \mu) \leq away(\hat{q}, \mu) = \hat{q} = \hat{z}.$$

Since $x > q - 2^{expo(x)-\mu}$, $away(x, \mu) \geq near(x, \mu) \geq \hat{q}$ by Lemma 2.28.

Case 3: $rc = \texttt{RC-NEAR}$ and $q\text{-}guard = 0$

Here, $\hat{q}$ is μ-exact, $rnd(x, rc, pc) = near(x, \mu)$, and $\hat{z} = trunc(\hat{q}, \mu) = \hat{q}$.

Since $x < \hat{q} + 2^{expo(\hat{q})-\mu}$ implies $near(x, \mu) \leq \hat{q} = \hat{z}$ by Lemma 2.28(b). But since $x > \hat{q} - 2^{expo(x)-\mu}$, $near(x, \mu) \geq \hat{q}$ by Lemma 2.28(c).

Case 4: $rc = \texttt{RC-NEAR}$ and $q\text{-}guard = 1$

In this case, $\hat{q}$ is $(\mu + 1)$-exact but not μ-exact. Let $a = q - 2^{expo(\hat{q})-\mu}$ and $b = q + 2^{expo(\hat{q})-\mu}$. By Lemma 2.27, $a = trunc(\hat{q}, \mu)$ and $b = away(\hat{q}, \mu)$.

Subcase 4.1: $rem\text{-}pos = 1$

In this case, $\hat{z} = b$ and $\hat{q} < x$. Since $x < \hat{q} + 2^{expo(\hat{q})-\mu} = b$,

$$near(x, \mu) \leq near(b, \mu) = b = \hat{z}.$$

But $x > q = b - 2^{expo(\hat{q})-\mu} \geq b - 2^{expo(x)-\mu}$, hence $near(x, \mu) \geq b$.

Subcase 4.2: $rem\text{-}neg = 1$

In this case, $\hat{z} = trunc(\hat{q}, \mu) = a$ and $x < \hat{q}$, hence $near(x, \mu) \leq a = \hat{z}$ by Lemma 2.28, and $x > q - 2^{expo(\hat{q})-\mu} = a$ implies $near(x, \mu) \geq near(a, \mu) = a$.

Subcase 4.3: $rem\text{-}zero = 1$

Here, $x = \hat{q}$, hence $near(x, \mu) = near(\hat{q}, \mu)$. We shall show $near(\hat{q}, \mu) = \hat{z}$. Note that by Lemma 2.29, $near(\hat{q}, \mu)$ is $(\mu - 1)$-exact.

If $q\text{-}lsb = 1$, then $\hat{z} = b$ and $a = trunc(\hat{q}, \mu)$ is not $(\mu - 1)$-exact by Lemma 4.14. Thus, $near(\hat{q}, \mu) \neq a$, which implies $near(\hat{q}, \mu) = b = \hat{z}$.

If $q\text{-}lsb = 0$, then $\hat{z} = a$, a is $(\mu - 1)$-exact by Lemma 4.14. It follows that b is not $(\mu - 1)$-exact, and hence $near(\hat{q}, \mu) = a$. □

We may now state the correctness theorem for division. Note that the bound on $expo(\hat{b})$ is required by Lemma 4.4 and is therefore unavoidable. The other constraint states that $expo(\hat{a}/\hat{b})$ may not assume either of the limiting values $1 - 2^{17}$ and 2^{17}. This is acceptable since the hardware would never be expected to return a value with either of those exponents. In particular, IEEE compliance only involves exponents that are accommodated by the 80-bit $(64, 15)$ format.

Theorem 2 *Assume* $op =$ `OP-DIV`, *rc is a rounding control specifier, pc is an external precision control specifier, and a and b are normal encodings such that* $expo(\hat{b}) \leq 2^{17} - 2$ *and* $2 - 2^{17} \leq expo(\hat{a}/\hat{b}) \leq 2^{17} - 1$. *Then z is a normal encoding and*

$$\hat{z} = rnd(\hat{a}/\hat{b}, rc, pc).$$

Proof: By the same reasoning that was used in the proof of Theorem 1, we may assume that $\hat{a} > 0$ and $\hat{b} > 0$. We need only show that the hypotheses of Lemma 4.15 are satisfied by $x = \hat{a}/\hat{b}$.

First note that our hypothesis regarding $expo(\hat{a}/\hat{b})$ yields the bounds on $|\hat{a}/\hat{b}|$ that are required by Lemma 4.9, which implies that $\hat{q}$ is $(\mu + 1)$-exact and

$$|\hat{q} - \hat{a}/\hat{b}| < 2^{min(expo(\hat{q}), expo(\hat{a}/\hat{b}))-\mu}.$$

This in turn implies the bounds on $\hat{q}$ that are required by Lemma 4.15, as well as $\hat{q} > 0$, and hence $get\text{-}sign(q) = sign = 0$.

Next, we apply Lemma 3.8 with $x = b$, $y = q$, $d = a$, and $z = rem$, which implies that

$$|\hat{a}/\hat{b}| > |\hat{q}| \Leftrightarrow |\hat{b}\hat{q}| < |\hat{a}| \Leftrightarrow get\text{-}man(rem)[M-2] = 1 \Leftrightarrow rem\text{-}pos = 1$$

and

$$\hat{a}/\hat{b} = \hat{q} \Leftrightarrow \hat{b}\hat{q} = \hat{a} \Leftrightarrow get\text{-}man(rem)[M-2:0] = inexact = 0 \Leftrightarrow rem\text{-}zero = 1.$$

But since exactly one of *rem-pos*, *rem-zero*, and *rem-neg* is nonzero, it follows that

$$|\hat{a}/\hat{b}| < |\hat{q}| \Leftrightarrow rem\text{-}neg = 1,$$

and all hypotheses of Lemma 4.15 are satisfied. □

In order to prove our correctness result for square root, a modification of Lemma 4.16 will be required:

Lemma 4.16 *Let* $\mu = mbits(pc)$. *Suppose* q *is normal,* $\hat{q}$ *is* $(\mu+1)$*-exact,* $sign = 0$*, and* $2^{1-2^{17}} < \hat{q} < 2^{2^{17}}(2-2^{1-\mu})$. *Let* $\ell, h \in \mathbb{Q}$ *such that*

(a) $\ell - 2^{min(expo(\hat{q}),expo(\ell))-\mu} < \hat{q} < h + 2^{min(expo(\hat{q}),expo(h))-\mu}$;

(b) if $rem\text{-}neg = 1$*, then* $\hat{q} > \ell$;

(c) if $rem\text{-}pos = 1$*, then* $\hat{q} < h$;

(d) if $rem\text{-}zero = 1$*, then* $\ell \leq \hat{q} \leq h$.

Then z *is normal and* $rnd(\ell, rc, pc) \leq \hat{z} \leq rnd(h, rc, pc)$.

Proof: We shall prove the first inequality; the proof of the second is similar.

Case 1: $rem\text{-}neg = 1$

Since $\ell < \hat{q}$, we may find x such that $\ell < x < \hat{q}$ and $x > \hat{q} - 2^{min(expo(\hat{q}),expo(x))-\mu}$. Then $rnd(\ell, rc, pc) \leq rnd(x, rc, pc)$, but by Lemma 4.15, $rnd(x, rc, pc) = \hat{z}$.

Case 2: $rem\text{-}pos = 1$

Choose x so that $\hat{q} < x < q + 2^{min(expo(\hat{q}),expo(x))-\mu}$ and $x > \ell$. Then $rnd(\ell, rc, pc) \leq rnd(x, rc, pc)$, but by Lemma 4.15, $rnd(x, rc, pc) = \hat{z}$.

Case 3: $rem\text{-}zero = 1$

Let $x = \hat{q}$. Then $\ell \leq x$, hence $rnd(\ell, rc, pc) \leq rnd(x, rc, pc)$, but by Lemma 4.15, $rnd(x, rc, pc) = \hat{z}$. □

Theorem 3 *Assume* $op =$ `OP-SQRT`*,* rc *is a rounding control specifier,* pc *is an external precision control specifier, and* b *is a normal encoding such that* $expo(\hat{b}) \leq 2^{17} - 2$. *Let* $\ell, h \in \mathbb{Q}$ *such that* $0 \leq \ell \leq h$ *and* $\ell^2 \leq \hat{b} \leq h^2$. *Then* z *is a normal encoding and*

$$rnd(\ell, rc, pc) \leq \hat{z} \leq rnd(h, rc, pc).$$

Proof: It suffices to show that the hypotheses of Lemmas 4.16 are satisfied. First, by Lemma 4.13, $\hat{q}$ is $(\mu+1)$-exact,

$$\ell < \hat{q} + 2^{min(expo(\hat{q}),expo(\ell))-\mu},$$

and

$$h > \hat{q} - 2^{min(expo(\hat{q}),expo(h))-\mu}.$$

Substituting $2^{\lfloor expo(\hat{b})/2 \rfloor}$ for ℓ in the same lemma, we have

$$\hat{q} > 2^{\lfloor expo(\hat{b})/2 \rfloor} - 2^{\lfloor expo(\hat{b})/2 \rfloor - \mu} > 2^{\lfloor expo(\hat{b})/2 \rfloor - 1} > 0,$$

hence

$$get\text{-}sign(q) = 0 = sign.$$

Similarly, substituting $2^{\lfloor expo(\hat{b})/2 \rfloor + 1}$ for h yields

$$\hat{q} < 2^{\lfloor expo(\hat{b})/2 \rfloor + 1} + 2^{\lfloor expo(\hat{b})/2 \rfloor + 1 - \mu} < 2^{\lfloor expo(\hat{b})/2 \rfloor + 2}.$$

Thus,

$$2^{2-2^{17}} < 2^{\lfloor expo(\hat{b})/2 \rfloor - 1} \leq \hat{q} < 2^{\lfloor expo(\hat{b})/2 \rfloor + 2} < 2^{2^{17}}.$$

Finally, we apply Lemma 3.8 with $x = y = q$, $d = b$, and $z = rem$, which yields the following:

(1) if $rem\text{-}neg = 1$, then $\hat{q}^2 > \hat{b} \geq \ell^2$, hence $\hat{q} > \ell$;

(2) if $rem\text{-}pos = 1$, then $\hat{q}^2 < \hat{b} \leq h^2$, hence $\hat{q} < h$;

(3) if $rem\text{-}zero = 1$, then $\hat{q}^2 = \hat{b}$, hence $\ell \leq \hat{q} \leq h$.

Thus, all hypotheses of Lemmas 4.16 are satisfied. □

5 Conclusion

As noted in the introduction, the practical value of formal verification has been illustrated in this exercise by the detection of two design flaws. Both of these were in the definition of the procedure *FPU-MUL*, but neither affected the results of multiplication. One was an error in the specification of the parameter r in the rare case in which *overflow* = 0 and *round-carryout-no-overflow* = 1, which would inevitably have led to erroneous quotients and square roots for certain inputs. The other was in the calculation of z in the *OP-BACK* case, and might have led to improper rounding of square roots, although we were unable to exhibit a concrete example of this behavior. It was not surprising that neither problem was exposed by traditional testing methods. Once they were identified, however, both were easily corrected before the design was committed to silicon.

Aside from the correction of errors, formal analysis may also provide insight that allows improvements in the efficiency of a design. For example, while the multiplier that was originally presented to us had a width of 76 bits, we were able to show, by representing it as an indefinite parameter M, that this width could effectively be reduced to 75 bits without sacrificing the accuracy of any of the operations that the multiplier supports.

Although the functionality of a physical device cannot be absolutely guaranteed by the properties of a mathematical model, a realistic model can provide a fairly high level of confidence. In this case, our analysis was based on a register-transfer model, far less abstract than the hardware models that are typically used in formal verification of floating point algorithms. It must be noted, however, that the evidentiality of our mechanical verification depends on the accuracy of several stages of manual translation. The original C encoding of the design was translated by hand into a special-purpose hardware description language, from which a gate-level implementation was eventually constructed. Meanwhile, our verification began with the pseudocode representation of the C program on which the lemmas and theorems of this paper are based. After detailed proofs of all of these results were derived informally (and this paper was essentially written), the pseudocode was translated into ACL2 along with the lemma statements. Finally, formal proofs of these statements were generated mechanically by guiding the ACL2 prover through each step of the informal proofs.

Obviously, our confidence in the final product would be enhanced if we could eliminate or mechanize any of the steps in these translations. This has been a focus of our more recent work: we have implemented a mechanical translator from AMD's hardware description language directly to the logic of ACL2, thereby reducing the possibility of human error in the formalization of hardware designs. In a report that is yet to be released, we describe the use of this translator in the mechanical verification of the AMD-K7 floating point adder.

Of course, a successful formal verification project requires a significant investment. The cost to AMD of the results presented here was five months of the author's time, divided approximately equally between writing the informal proofs and checking them mechanically. Much of this time, however, was spent developing general methods and results, especially the theory of floating point arithmetic presented in Section 2, which could be reused in any floating point verification effort. We have already applied the same results to several problems, and it is our hope that others will find them useful in similar projects. Thus, the ACL2 formalization of this theory has been made available through the ACL2 Web site.[1]

References

[1] ACL2 Web site, `http://www.cs.utexas.edu/users/moore/acl2/`.

[2] Anderson, S.F., Earle, J.G., Goldschmidt, R.E., and Powers, D.M., "The IBM System/360 Model 91 Floating Point Execution Unit", *IBM Journal of Research and Development*, 11:34-53, January, 1967.

[3] Bryant, R.E., "Verification of Arithmetic Functions with Binary Moment Diagrams", Technical Report CMU-CS-94-160, School of Computer Science, Carnegie-Mellon University, 1994.

[4] Clarke, E.M. and Zhao, X., "Word Level Symbolic Model Checking: A New Approach for Verifying Arithmetic Circuits", Technical Report CMU-CS-95-161, School of Computer Science, Carnegie-Mellon University, 1995.

[5] Institute of Electrical and Electronic Engineers, "IEEE Standard for Binary Floating Point Arithmetic", Std. 754-1985, New York, NY, 1985.

[6] Moore, J, Lynch, T., and Kaufmann, M., "A Mechanically Checked Proof of the Correctness of the Kernel of the $AMD5_K86$ Floating Point Division Algorithm", *IEEE Transactions on Computers*, 47:9, September, 1998.

[7] Oberman, S.F., "Division and Square Root for the AMD-K7 FPU", Advanced Micro Devices, Milpitas, CA, March, 1997.

[8] Russinoff, D.M., "A Mechanically Checked Proof of IEEE Compliance of the AMD-K5 Floating Point Square Root Microcode", to appear in *Formal Methods in System Design*, 1998.
`http://www.russinoff.com/papers/fsqrt.html`.

[9] Steele, G.L., Jr., *Common Lisp The Language*, 2nd edition, Digital Press, 1990.