Design and correctness proof of an emulation of the floating-point operations of the Electrologica X8 : a case study

Design and correctness proof of an emulation of the floating–point operations of the Electrologica X8. A case study

F.E.J. Kruseman Aretz

March 30, 2010

1 Introduction

Some time ago I decided to write an emulator for a Dutch computer from the sixties of the previous century, the Electrologica X8, in order to be able to run its ALGOL 60 implementation and to do some measurements with it. That emulator was written in (standard ISO) Pascal.

Part of it was the emulation of its floating–point operations. I started by designing them directly in Pascal, but then immediately encountered a number of complications.

First of all the fact that, using integer arithmetic of restricted capacity, the mantissa of 40 bits had to be divided over two 32–bit integer variables. But also the limitations on the value of the (binary) exponent and, more generally, the treatment of all exceptional cases caused much troubles.

Moreover, I wished to emulate not only the operations themselves, but also their timing, in order to be able to compare X8 program execution speed with that of other computers. However, I did not have any documentation of the hardware implementation. I remembered only some details, especially some of the tricks to improve the efficiency. I tried to reflect in my code a possible structure of the hardware, incorporating those details. Altogether a complicated affair.

Therefore, I decided to express the logic of the emulation in terms of guarded command language[1, 2] first, without any capacity restrictions for the exponent. The routines for addition, multiplication, and division were designed using operational arguments. I felt, then, a need to prove them correct. And not without results: a correction of a constant in the routine for addition (from 81 to 80) showed necessary to complete its proof! Having the logic of the routines available it was not too difficult to design the Pascal version with all its nasty details.

For the correctness proofs I used weakest–precondition logic[4].

These proofs were more complex and and more lengthy than I expected. After completion I asked the Eindhoven Tuesday Afternoon Club for help, in the hope that a simplification would be possible. They looked, in my presence, into it one afternoon, but didn't get any

further than a few lines of the first proof and their remarks were not very helpful. They returned never to the subject after that session.

The structure of this report is as follows.

In Section 2, I give the necessary information about the Electrologica X8 and its floating–point representation. Moreover the rounding of a real number to a floating–point number is defined and some useful properties thereof are mentioned.

Sections 3, 4, and 5 present the guarded–command versions of multiplication, division, and addition, respectively. Each of them has four subsections. First, the operational design considerations are presented. Second, the resulting procedure is given; it follows the design considerations faithfully and is, hopefully, comprehensible without further comments or assertions. Third, this procedure is given again, part by part, now augmented with the assertions and invariants necessary for a formal correctness proof. I also give here comments on their choice, some of the resulting proof obligations, and some critical ingredients of the proofs. The proof obligations and, where necessary, hints for the proofs are given in Appendix A. A fourth subsection presents some final remarks.

In Section 6 I try to sum up some of my experiences.

2 The Electrologica X8

Electrologica was a Dutch computer factory, founded in 1956. It produced the Electrologica X1 (from 1958) and its successor, the Electrologica X8 (from 1965). The latter was more or less upwards compatible with the former, about a factor of 12 faster, and in addition it had floating–point hardware: an additional register **F** and instructions for floating–point addition, subtraction, multiplication, and division.

Floating–point numbers in **F** were represented by the Grau–representation [3]. In that representation integral values are a natural subset of the floating–point numbers. Hence register **F** could be used for both integer and real–number arithmetic, useful for the implementation of the mixed–mode arithmetic expression of ALGOL 60.

2.1 Floating–point numbers

In the Grau representation a number is characterized by an integral mantissa and an integral exponent. The X8 used 40 bits to represent the absolute value of the mantissa and a separate bit for the number sign. The binary exponent was encoded in 12 bits (including its sign), but in this report we will not limit its value.

So let us introduce the set of floating–point numbers $\mathcal{F}$ as

$$\mathcal{F} = \{(s, m, e) \mid s \in \{+1, -1\} \wedge m \in \mathcal{N} \wedge 0 \leq m < 2^{40} \wedge e \in \mathcal{Z}\}$$

For $f \in \mathcal{F}$ we denote its components by $f.s$, $f.m$, and $f.e$, so $f = (f.s, f.m, f.e)$.

Let $f = (s, m, e)$ with $f \in \mathcal{F}$. It represents a real number

$$val.f = s \times m \times 2^e$$

In general, these representations are not unique. If $m < 2^{39}$, we can double m, at the same time decreasing e by 1. If m is even, we can half m, at the same time increasing e by 1. Two floating–point numbers f_1 and f_2 are called equivalent, denoted $f_1 \cong f_2$, if they represent the same real number:

$$(f_1 \cong f_2) = (val.f_1 = val.f_2)$$

Hence

$$(s_1, m_1, e_1) \cong (s_2, m_2, e_2) = (s1 \times m1 \times 2^{e1} = s2 \times m2 \times 2^{e2})$$

The standard representation in the X8 is the one for which $|e|$ is minimal (closest to zero). Consequently, all integral values n with $|n| < 2^{40}$ will be represented preferably by a floating–point number $(sign.n, |n|, 0)$.

All floating–point operations of the X8 accepted operands that were not in standard form and delivered the result in register **F** in standard form.

The following algorithm brings a floating–point number (s, m, e) into standard form:

procedure standardize((s,m,e): f_number);
{*brings* (s, m, e) *into standard form*}
begin $\{(s, m, e) = (ss, mm, ee)\}$
  if m $= 0 \rightarrow$ s, e $:= +1, 0$
  [] m $> 0 \rightarrow$ **do** $\{$*invariant* $(s, m, e) \in \mathcal{F} \wedge (s, m, e) \cong (ss, mm, ee)\}$
      even.m $\wedge$ e $< 0 \rightarrow$ m, e $:=$ m **div** 2, e $+ 1$
    [] m $< 2^{39} \wedge$ e $> 0 \rightarrow$ m, e $:=$ m $*$ 2, e $- 1$
    od
  fi $\{(s, m, e) \in \mathcal{F} \wedge (s, m, e) \cong (ss, mm, ee) \wedge (s, m, e)$ *in standard form*$\}$
end;

We are now in the position to formulate the quality requirement that was fulfilled by the X8 hardware: the floating–point operations '+', '−', '∗', and '/' all delivered the best possible result, i.e. that floating–point number in standard form whose value is closest to the exact result. In case that the exact result of the operation was precisely midway two consecutive floating–point numbers, the result was rounded upwards for positive results and downwards for negative results. This uniquely specifies the functionality of these operations.

In our emulations of these operations we will stick to this quality requirement.

The X8 floating–point numbers can be partitioned into intervals. In each interval the numbers are equidistant and their values run from $2^{39} \times 2^e, (2^{39} + 1) \times 2^e, \cdots, 2^{40} \times 2^e$ with an inter-spacing of 2^e (for fixed value of e). At the border of consecutive intervals that inter-spacing changes by a factor of 2.

This has consequences for the formal expression of our quality requirement if the exact result of an operation is near the boundary between two successive intervals:

$2^{39} + 0.4$ rounds to 2^{39}, i.e., floating–point number $(+1, 2^{39}, 0)$,
$2^{39} - 0.4$ rounds to $2^{39} - 0.5$, i.e., floating–point number $(+1, 2^{40} - 1, -1)$,

the first value lying in an interval with inter-spacing 1, the second in an interval with inter-spacing 0.5.

2.2 Rounding

We define a function rnd from $\Re$ to $\mathcal{F}$ in the following way.

for $x > 0$: $rnd.x = (+1, m, e)$ where $2^{39} \leq m < 2^{40}$
and if $m > 2^{39}$ then $(m - 1/2) \times 2^e \leq x < (m + 1/2) \times 2^e$
and if $m = 2^{39}$ then $(m - 1/4) \times 2^e \leq x < (m + 1/2) \times 2^e$

for $x = 0$: $rnd.0 = (+1, 0, 0)$

for $x < 0$: $rnd.x = (-1, m, e)$ where $(+1, m, e) = rnd.(-x)$

Note that $rnd.x$ is, in general, not in standard form.

The quality requirement for the floating–point operations that was formulated in the previous subsection can be formally expressed in terms of rounding. For multiplication it reads:

$$f_1 * f_2 \cong rnd.(val.f_1 \times val.f_2)$$

with $f_1 * f_2$ in standard form[1].

We mention, without proof, the following properties of rnd:

1. $rnd.x \in \mathcal{F}$ for all $x \in \Re$.
2. in $rnd.x = (s, m, e)$ all of s, m, and e are uniquely determined.
3. rnd is monotonically non–decreasing, i.e.
   $(\forall x, x' : x, x' \in \Re : (x < x') \Rightarrow (val.rnd.x \leq val.rnd.x'))$
4. $rnd.val.f \cong f$ for $f \in \mathcal{F}$. Hence:
5. rnd is idempotent, i.e. $rnd.val.rnd.x = rnd.x$ for $x \in \Re$.
6. if $rnd.x = (s, m, e)$ then $rnd.(x \times 2^{ee}) = (s, m, e + ee)$, for all $x \in \Re$ with $x \neq 0$.
7. only 0 maps to $(+1, 0, 0)$, i.e. if $rnd.x = (+1, 0, 0)$ then $x = 0$.
8. $(\forall x, z : (x \in \Re) \wedge (z \in \mathcal{F}) :$
   $(|val.z - x| \geq |val.rnd.x - x|) \wedge$
   $((|val.z - x| = |val.rnd.x - x|) \Rightarrow (|val.rnd.x| \geq |val.z|))$.

The last property above expresses that for $x \in \Re$, $rnd.x$ is indeed a best approximation of x in $\mathcal{F}$ and that, in case x lies midway two consecutive $\mathcal{F}$–members, the one greatest in absolute value is taken for $rnd.x$.

We prove the following two properties:

[1] In the sequel we will abbreviate $val.f_1 \times val.f_2$ to $f1 \times f2$. The same holds for the operators / and +. In that notation the quality requirement given above reads:

$$f_1 * f_2 \cong rnd.(f_1 \times f_2) \wedge f_1 * f_2 \text{ in standard form.}$$

1. Let $x > 0$ and $y = val.rnd.x$.
   Then $\left|\frac{y-x}{y}\right| \leq 2^{-40}$.

   Proof:
   Let $rnd.x = (+1, m, e)$. Then $y = m \times 2^e$ with $2^{39} \leq m < 2^{40}$.
   From the definition of rnd we have $|y - x| \leq 1/2 \times 2^e$.
   We derive

$$\begin{array}{ll} & \left|\frac{y-x}{y}\right| \\ \leq & (1/2 \times 2^e)/(m \times 2^e) \\ = & 1/(2 \times m) \\ \leq & 1/(2 \times 2^{39}) \\ = & 2^{-40} \end{array}$$

   □

2. Let $y = (+1, m, e)$ be $\in \mathcal{F}$ with $val.y > 0$ and x be $\in \Re$ such that $|x| < val.y \times 2^{-41}$.
   Then $rnd.(val.y + x) \cong y$.

   Proof:
   For $x = 0$ the proof is trivial: $rnd.val.y \cong y$, for all $y \in \mathcal{F}$.
   So let $x \neq 0$. Without loose of generality we may assume $2^{39} \leq m < 2^{40}$.
   If $x > 0$ then

$$\begin{array}{ll} & (m - 1/2) \times 2^e \\ < & (m - 1/40) \times 2^e \\ < & val.y \\ < & val.y + x \\ < & (m + m \times 2^{-41}) \times 2^e \\ < & (m + 2^{40} \times 2^{-41}) \times 2^e \\ = & (m + 1/2) \times 2^e, \end{array}$$

   hence $rnd.(val.y + x) = (+1, m, e)$.
   If $x < 0$ then

$$\begin{array}{ll} & (m - 1/2) \times 2^e \\ = & (m - 2^{40} \times 2^{-41}) \times 2^e \\ < & (m - m \times 2^{-41}) \times 2^e \\ < & val.y + x \\ < & val.y \\ < & (m + 1/2) * 2^e, \end{array}$$

   whereas for $m = 2^{39}$ even $(m - 1/4) \times 2^e < val.y + x$.

   □

3 Multiplication

As mentioned in the introduction this section consists of four subsections:

1. operational design considerations,
2. the resulting algorithm,
3. the correctness proof of the algorithm, and
4. some final remarks.

The algorithm follows the design considerations faithfully and should be comprehensible without further comments or assertions. It is, therefore, presented without intermediate assertions. These are given, in full detail, in the correctness proof of the algorithm.

3.1 Operational design considerations

We start the description of the emulation of the floating–point operations of the X8 with that of multiplication. The task is simple, at least in principle: multiply the two mantissas, add the two binary exponents, and determine the sign according to the standard rule. Complications arise only if the product of the two mantissas exceeds the maximum value of $2^{40} - 1$: then it has to be brought back within the capacity by (successive) halving the mantissa and incrementing the exponent. Moreover we have to carry out a proper round–off.

There are, however, some good reasons to do it differently. In the first place we expressed already our desire to make an emulation that reflects more or less what we know about the original hardware. In the sixties of the previous century, registers and accumulator logic were rather expensive, with a price proportional to their bit length. It was, therefore, prohibitive to accommodate an 80 bit product, of which at most 41 bits were of interest: 40 bits for the representation of the mantissa of the result and 1 bit for the rounding information. Secondly, also in our emulation it is not attractive to manipulate integer values of 80 bits (which, with an integer capacity of 31 bits, must be represented by at least three integer variables). In the third place we remember that in the X8 hardware small integer factors were dealt with rather efficiently: the execution time of multiplication depended on the number of relevant bits of the operands.

Therefore, we took the following approach:
We build the mantissa as a 42–bit[2] integer m, starting by 0. We scan iteratively the bits of the multiplier, from right to left. For each non–zero bit of the multiplier, we add the multiplicand to m. In principle, we multiply the multiplicand by 2 for each (zero or non–zero) bit of the multiplier. If thereby, however, the multiplicand would exceed a length of 42 bits, we half m instead, incrementing the binary exponent of the product by 1 at the same time.

We can express this approach by the piece of (pseudo) code given in the next subsection.

[2]In Section 3.3 we show that a 41–bit integer would not suffice.

3.2 The resulting procedure

procedure f_multiply(f2: f_number);
{*computes, for global* $f1$, $f1 := f1 * f2$}
var m, e, guard: integer; s: sign;
begin $\{f1 = g1 \wedge f2 = g2\}$
  let f1, f2 = (s1,m1,e1), (s2,m2,e2);
  m, e := 0, 0;
{*multiply mantissa's:*}
  do m2 > 0 →
    if odd.m2 → m, m2 := m + m1, m2 − 1
    [] even.m2 → **skip**
    fi;
    if m1 < 2^{41} → m1 := 2 ∗ m1
    [] m1 ≥ 2^{41} → m, e := m **div** 2, e + 1
    fi;
    m2 := m2 **div** 2
  od;
{*prepare resulting mantissa for proper round–off:*}
  guard := 0;
  do m ≥ 2^{40} → guard, m, e := m **mod** 2, m **div** 2, e + 1
  od;
{*round:*}
  if guard = 1 **and** m = $2^{40} - 1$ → m, e := 2^{39}, e + 1
  [] guard = 1 **and** m < $2^{40} - 1$ → m := m + 1
  [] guard = 0 → **skip**
  fi;
{*form result:*}
  if s1 = s2 → s := +1
  [] s1 ≠ s2 → s := −1
  fi;
  f1 := (s,m,e + e1 + e2);
  standardize(f1)
$\{f1 \cong rnd.(g1 \times g2) \wedge f1$ *in standard form*$\}$
end {*f_multiply*};

3.3 Correctness proof of f_multiply

The following assertion holds, and is used, throughout the proof; all assertions should be augmented by it:

$$P = 0 \leq g1.m < 2^{40} \wedge 0 \leq g2.m < 2^{40} \wedge 0 \leq m1 < 2^{42} \wedge 0 \leq m2$$

We start by proving the correctness of the code for multiplying the two mantissa's. We do so only for the case that $g2.m > 0$: invariant P_2 holds trivially if $g2.m = 0$. The detailed proof obligations are worked out in Appendix 1.

$\{f1 = g1 \wedge f2 = g2 \wedge g2.m > 0\}$
m, e := 0, 0;
$\{I_0\}$
do m2 $> 0 \rightarrow$
  $\{m2 > 0 \wedge I_0\}$
  if odd.m2 $\rightarrow$ m, m2 := m + m1, m2 − 1
  [] even.m2 $\rightarrow$ **skip**
  fi;
  $\{P_0\}$
  if m1 $< 2^{41} \rightarrow$ m1 := 2 ∗ m1
  [] m1 $\geq 2^{41} \rightarrow$ m, e := m **div** 2, e + 1
  fi;
  $\{P_1\}$
  m2 := m2 **div** 2
  $\{I_0\}$
od;
$\{P_2\}$

where

$$\begin{aligned} P_2 = \;& 0 \leq m < 2^{42} \wedge \\ & 0 \leq e \wedge \\ & (m < 2^{40} \rightarrow e = 0) \wedge \\ & m \times 2^e \leq g1.m \times g2.m < (m+1) \times 2^e \end{aligned}$$

In P_2 the clause '$m < 2^{40} \rightarrow e = 0$' might seem superfluous for the correctness proof of the multiplication. Note, however, that $P_2 \wedge e = 0$ implies $m = g1.m \times g2.m$, since both m and $g1.m \times g2.m$ are integral. This fact is used in the last step of the correctness proof. The clause '$0 \leq e$' is required in a later stage of the proof to show that incrementation of e by 1 will lead to a non–zero value of e.

The invariant for the iteration, I_0, reads:

$$\begin{aligned} I_0 = \;& 0 \leq m \leq m1 \wedge \\ & 0 \leq e \wedge \\ & (m1 < 2^{41} \rightarrow e = 0) \wedge \\ & (m2 > 0 \vee m1 \div 2 \leq m) \wedge \\ & (m + m1 \times m2) \times 2^e \leq g1.m \times g2.m < (m + 1 + m1 \times m2) \times 2^e \end{aligned}$$

The first assertion of I_0 is used to prove the first assertion of P_0, the third and fourth assertions of I_0 are necessary to prove the third assertion of P_0.

Proof obligations here are the validity of I_0 at the start of the iteration, the invariance of

I_0 in the iteration, and the fact that $I_0 \wedge m2 = 0$ implies P_2.

At the start we have[3]:

$(P \wedge f1 = g1 \wedge f2 = g2 \wedge g2.m > 0)$
$\Rightarrow$
$(P \wedge I_0)\,[m/0, e/0]$

which holds since $f1 = g1 \wedge f2 = g2$ implies $m1 = g1.m \wedge m2 = g2.m$. Consequently $g2.m > 0$ implies $m2 > 0$, hence $m2 > 0 \vee m1 \div 2 \leq m$.

At the end of the iteration we have $I_0 \wedge m2 = 0$. Hence P_2, since for $g2.m > 0$ we derive:

$$\begin{array}{ll} & m2 = 0 \wedge m < 2^{40} \\ \Rightarrow & m1 \div 2 \leq m \wedge m < 2^{40} \\ \Rightarrow & m1 \div 2 < 2^{40} \\ \Rightarrow & m1 \div 2 \leq 2^{40} - 1 \\ \Rightarrow & m1 \leq 2 \times (2^{40} - 1) + 1 \\ \Rightarrow & m1 \leq 2^{41} - 1 \\ \Rightarrow & m1 < 2^{41} \\ \Rightarrow & e = 0 \end{array}$$

Of course we have to prove that $\{I_0 \wedge m2 > 0\} S \{I_0\}$, where S is the controlled statement, showing that I_0 is an invariant indeed. There are three steps:

1. $\{I_0 \wedge m2 > 0\}$
   if odd.m2 → m, m2 := m + m1, m2 − 1
   [] even.m2 → **skip**
   fi;
   $\{P_0\}$

   where P_0 reads:

   $$\begin{array}{rl} P_0 = & 0 \leq m \leq 2 \times m1 \wedge \\ & 0 \leq e \wedge \\ & (m1 < 2^{41} \rightarrow e = 0) \wedge \\ & (m2 > 0 \vee m1 \leq m) \wedge \\ & (m + m1 \times m2) \times 2^e \leq g1.m \times g2.m < (m + 1 + m1 \times m2) \times 2^e \wedge \\ & even.m2 \end{array}$$

   The proof obligations read:
   $(I_0 \wedge m2 > 0 \wedge odd.m2) \Rightarrow P_0\,[m/m + m1, m2/m2 - 1]$ and
   $(I_0 \wedge m2 > 0 \wedge even.m2) \Rightarrow P_0$.

   The proofs are elementary, since assignment 'm, m2 := m + m1, m2 − 1' leaves term $m + m1 \times m2$ invariant.

2. $\{P_0\}$
   if m1 $< 2^{41}$ → m1 := 2 ∗ m1
   [] m1 $\geq 2^{41}$ → m, e := m **div** 2, e + 1
   fi;
   $\{P_1\}$

   where P_1 reads:

[3] notation $Q\,[x/y]$ is used for assertion Q with expression y substituted for identifier x.

$$
\begin{aligned}
P_1 = \;& 0 \le m \le m1 \wedge \\
& 0 \le e \wedge \\
& (m1 < 2^{41} \rightarrow e = 0) \wedge \\
& (m2 > 0 \vee m1 \div 2 \le m) \wedge \\
& (m + m1 \times (m2 \div 2)) \times 2^e \le g1.m \times g2.m < \\
& \quad (m + 1 + m1 \times (m2 \div 2)) \times 2^e \wedge \\
& even.m2
\end{aligned}
$$

The proof obligations read:
$(P_0 \wedge m1 < 2^{41}) \Rightarrow P_1\,[m1/2 \times m1]$ and
$(P_0 \wedge m1 \ge 2^{41} \Rightarrow P_1\,[m/m \div 2, e/e + 1]$.

For the proof we use that $m - 1 \le 2 \times (m \div 2) \le m$ and that $even.m2$ implies $2 \times (m2 \div 2) = m2$, from which it follows that

$$
\begin{aligned}
& (m \div 2 + m1 \times (m2 \div 2)) \times 2^{e+1} \\
\le\; & (m + m1 \times m2) \times 2^e \\
\le\; & g1.m \times g2.m \\
<\; & (m + 1 + m1 \times m2) \times 2^e \\
\le\; & (m \div 2 + 1 + m1 \times (m2 \div 2)) \times 2^{e+1}\ \}
\end{aligned}
$$

3. $\{P_1\}$
   m2 := m2 **div** 2
   $\{I_0\}$

   The proof obligation reads:
   $P_1 \Rightarrow I_0\,[m2/m2 \div 2]$.

   Since $even.m2 \wedge m2 \div 2 = 0$ implies $m2 = 0$, the third statement of the first iteration indeed restores invariant I_0.

We proceed by proving the next step of *f_multiply*:

$\{P_2\}$
guard := 0;
$\{I_1\}$
do m $\ge 2^{40} \rightarrow$ guard, m, e := m **mod** 2, m **div** 2, e + 1
od;
$\{P_3\}$

where

$$
\begin{aligned}
P_3 = \;& 0 \le m < 2^{40} \wedge \\
& 0 \le e \wedge \\
& guard \in [0, 1]) \wedge \\
& (m < 2^{39} \rightarrow e = 0) \wedge \\
& (e > 0 \vee (guard = 0 \wedge m = g1.m \times g2.m)) \wedge \\
& (m + guard/2) \times 2^e \le g1.m \times g2.m < (m + guard/2 + 1/2) \times 2^e
\end{aligned}
$$

Iteration invariant I_1 reads:

$$
\begin{aligned}
I_1 = \;& 0 \leq m < 2^{42} \wedge \\
& 0 \leq e \wedge \\
& \mathit{guard} \in [0,1]) \wedge \\
& (m < 2^{39} \rightarrow e = 0) \wedge \\
& (e > 0 \vee (\mathit{guard} = 0 \wedge m = g1.m \times g2.m)) \wedge \\
& (m + \mathit{guard}/2) \times 2^e \leq g1.m \times g2.m < (m+1) \times 2^e \wedge \\
& (m < 2^{40} \rightarrow g1.m \times g2.m < (m + \mathit{guard}/2 + 1/2) \times 2^e)
\end{aligned}
$$

The proof obligations read:

$P_2 \Rightarrow I_1\,[\mathit{guard}/0]$,
$(I_1 \wedge m \geq 2^{40}) \Rightarrow I_1\,[\mathit{guard}/m \textbf{ mod } 2, m/m \div 2, e/e+1]$, and
$(I_1 \wedge m < 2^{40}) \Rightarrow P_3$.

Here it is that we use our knowledge that $0 \leq e$ and therefore $\neg\,(e+1=0)$.

The last step in our proof is the demonstration of:

$\{P_3\}$
if guard = 1 **and** m = $2^{40} - 1 \rightarrow$ m, e := 2^{39}, e + 1
[] guard = 1 **and** m < $2^{40} - 1 \rightarrow$ m := m + 1
[] guard = 0 $\rightarrow$ **skip**
fi;
$\{R\}$

where assertion R is given by:

$$
\begin{aligned}
R = \;& 0 \leq m < 2^{40} \wedge \\
& 0 \leq e \wedge \\
& (m < 2^{39} \rightarrow e = 0) \wedge \\
& (e > 0 \vee m = g1.m \times g2.m) \wedge \\
& (m - 1/2) \times 2^e \leq g1.m \times g2.m < (m + 1/2) \times 2^e \wedge \\
& (m = 2^{39} \rightarrow (m - 1/4) \times 2^e \leq g1.m \times g2.m)
\end{aligned}
$$

This amounts to proving:

$(P_3 \wedge \mathit{guard} = 1 \wedge m = 2^{40} - 1) \Rightarrow R\,[m/2^{39}, e/e+1]$,
$(P_3 \wedge \mathit{guard} = 1 \wedge m < 2^{40} - 1) \Rightarrow R\,[m/m+1]$, and
$(P_3 \wedge \mathit{guard} = 0) \Rightarrow R$.

which is straightforward.

It follows from R straightforward that $(+1, m, e) \cong \mathit{rnd}.(g1.m \times g2.m)$, since for $m \geq 2^{39}$ even the stronger $(+1, m, e) = \mathit{rnd}.(g1.m \times g2.m)$ holds, whereas for $m < 2^{39}$ we have $(+1, g1.m \times g2.m, 0) \in \mathcal{F}$, hence $(+1, g1.m \times g2.m, 0) \cong \mathit{rnd}.(g1.m \times g2.m)$. This implies immediately our quality requirement for multiplication, reading

$$f1 \cong \mathit{rnd}.(g1 \times g2).$$

3.4 Final remarks

Apart from proving our quality requirement for multiplication, we also proved that whenever m has less then 40 bits, $m = g1.m \times g2.m$ due to the fact that, according to R,

$m < 2^{39}$ implies $e = 0$, which in turn implies $m = g1.m \times g2.m$. Hence in that case the product is exact: $val.f1 = g1 \times g2$.

This is an important aspect: it means that the floating–point multiplication can be used for integer arithmetic. If both operands are integers (i.e., $g1.e = g2.e = 0$) their product is integral and exact, provided that $g1.m \times g2.m < 2^{39}$.

For proving this, we had to incorporate in most assertions some additional clauses, which were not necessary for the proof of the quality requirement itself. It took some effort to find the clause

$g2.m > 0 \rightarrow (m2 > 0 \vee m1 \div 2 \leq m)$

in I_0, by which the clause

$m < 2^{40} \rightarrow e = 0$

in P_2 could be proved.

We also observe that if $val.g1 = 0$ or/and $val.g2 = 0$ then the result of their floating–point multiplication has zero value too. This follows from P_2:

$(P_2 \wedge g1.m \times g2.m = 0) \rightarrow m \times 2^e \leq 0$,

hence $m = 0$, which is retained in the remainder of the code..

An example in which in the rounding phase the first alternative is chosen is given by the multiplication of $2^{21} - 1$ by $2^{21} + 1$. The first phase of the multiplication leads to $m = 2^{41} - 1$ and $e = 1$ (showing that in P_2 a bound $g1.m \times g2.m < (m + 1/2) \times 2^e$ would have been too sharp), the second phase leads to $m = 2^{40} - 1$, $e = 2$, and $guard = 1$, and the rounding phase to $m = 2^{39}$ and $e = 3$. The best answer indeed, where $2^{42} - 1$ (the exact product) lies between $(2^{40} - 1) \times 2^2$ and $2^{39} \times 2^3$, but closer to the latter than to the former value.

The same example shows the need for 42 bits rather than 41 bit results in the multiplication phase (by allowing m to grow larger than $2^{41} - 1$ rather than restricting its value to at most $2^{41} - 1$). Otherwise the multiplication phase would lead to $m = 2^{40} - 1$ and $e = 2$, being at the same time the (erroneous) final result.

That in the multiplication phase m can grow beyond 2^{41} is shown by multiplying $2^{39} - 1$ by 5, resulting in $m = 2^{41} + 2^{39} - 5$, $e = 0$, $m1 = 2^{42} - 1$, and $m2 = 0$ (indeed both m and $m1$ below 2^{42}, as asserted in P_2 and P, respectively).

We see that for a small mantissa of multiplier $f2$ the number of additions and shifts is minimal: the multiplication of the mantissa's ends when the multiplier bits are exhausted. In the hardware, and also in the Pascal version of the emulation, multiplier and multiplicand were interchanged if the number of significant bits of the multiplier exceeded that of the multiplicand.

In the code for multiplying the two mantissas assignment 'm2 := m2 − 1' in

```
if odd.m2 → m, m2 := m + m1, m2 − 1
[] even.m2 → skip
fi
```

is superfluous: for odd values of $m2$ the value of $m2 \div 2$ is equal to that of $(m2 - 1) \div 2$. Omitting that assignment complicates, however, the assertions in the correctness proof tremendously.

In the code for multiplying the two mantissas the part:

$\{P_0\}$
if m1 $< 2^{41} \rightarrow$ m1 := m1 $*$ 2
[] m1 $\geq 2^{41} \rightarrow$ m, e := m **div** 2, e + 1
fi;
$\{P_1\}$
m2 := m2 **div** 2
$\{I_0\}$

can be rewritten as:

$\{P_0\}$
if m1 $< 2^{41} \rightarrow$ m1, m2 := m1 $*$ 2, m2 **div** 2
[] m1 $\geq 2^{41} \rightarrow$ m, e, m2 := m **div** 2, e + 1, m2 **div** 2
fi
$\{I_0\}$

complicating the code but simplifying the correctness proof. From an operational point of view the transformation from the latter to the former code is trivially correct, whereas the correctness proof by means of weakest preconditions requires the construction of assertion P_1. We will see more examples of this phenomenon.

4 Division

4.1 Operational design considerations

In the computation of $f1/f2$, we want, by shifting the mantissa's, reach the situation that $1 \leq f1.m/f2.m < 2$, by either doubling $f2.m$ until $2 \times f2.m$ exceeds $f1.m$ or doubling $f1.m$ as long as it is smaller than $f2.m$. In this way we prepare, in a minimal number of shift steps, the two operands for the division stage.

In this division stage, we build a quotient mantissa m of 40 bits in 40 iterations steps, starting with $m = 0$.

In each step we inspect whether we can subtract $f2.m$ from $f1.m$ without $f1.m$ becoming negative. If we can, we do so and add 1 to the quotient mantissa m. Moreover we double m before the inspection, and we double $f1.m$ after the operation. Thanks to the preparation prior to this iteration, in the first iteration step indeed $f1.m$ exceeds $f2.m$. Therefore we end, after 40 iteration steps, with a value of m fulfilling $2^{39} \leq m < 2^{40}$ and a division remainder in f1.m between 0 and $2 \times f2.m$, which is then used for properly rounding m.

We do not consider the exception $g2.m = 0$, i.e. division by zero.

4.2 The resulting procedure

```
procedure f_divide(f2: f_number);
{computes, for global f1, f1 := f1/f2}
var m, e, i, guard: integer; s: sign;
begin {f1 = g1 ∧ f2 = g2 ∧ g2.m > 0}
  let f1, f2 = (s1,m1,e1), (s2,m2,e2);
  if m1 = 0 → m, e := 0, 0
  [] m1 > 0 ∧ m2 > 0 →
      e := 0;
    {shift mantissa's:}
      do m1 ≥ 2 * m2 → m2, e := 2 * m2, e + 1
      [] m1 < m2 → m1, e := 2 * m1, e − 1
      od;
    {divide:}
      m, i := 0, 0;
      do i < 40 →
        if m1 ≥ m2 → m, m1, i := 2 * m + 1, 2 * (m1 − m2), i + 1
        [] m1 < m2 → m, m1, i := 2 * m, 2 * m1, i + 1
        fi
      od;
    {round:}
      if m1 ≥ m2 → m := m + 1
      [] m1 < m2 → skip
      fi;
      e := e − 39 + e1 − e2
  fi;
{form result:}
  if s1 = s2 → s := +1
  [] s1 ≠ s2 → s := −1
  fi;
  f1 := (s,m,e);
{f1 = rnd.(g1/g2)}
  standardize(f1)
{f1 ≅ rnd.(g1/g2) ∧ f1 in standard form}
end {f_divide};
```

4.3 Correctness proof of f_divide

In our proof we consider only the case that $g1.m > 0$. Again all proof obligations are worked out in Appendix A.

As was the case with multiplication, we have a general assertion by which all assertions in the proof should be augmented:

$$P = 1 \leq g1.m < 2^{40} \wedge 1 \leq g2.m < 2^{40} \wedge 0 \leq m1 < 2^{41} \wedge 1 \leq m2 < 2^{40}$$

We start by proving:

$\{f1 = g1 \wedge f2 = g2\}$
e := 0;
$\{I_0\}$
do m1 $\geq 2 *$ m2 $\rightarrow$ m2, e := $2 *$ m2, e + 1
[] m1 $<$ m2 $\rightarrow$ m1, e := $2 *$ m1, e $-$ 1
od;
$\{P_0\}$

Note that the iteration ends since both $m1 > 0$ and $m2 > 0$.

An easy and reasonable choice for P_0 would be:

$$P_0 = \begin{array}{l} 1 \leq m1/m2 < 2 \wedge \\ (m1/m2) \times 2^e = g1.m/g2.m \end{array}$$

with invariant I_0 reading:

$$I_0 = \begin{array}{l} (m1 < 2 \times m2 \vee m1 < 2^{40}) \wedge \\ (m1/m2) \times 2^e = g1.m/g2.m \end{array}$$

(the first clause of I_0 being necessary for proving that assignment 'm2 := $2 *$ m2' does not invalidate $m2 < 2^{40}$), but by this choice of P_0 we cannot infer that in the rounding phase of the algorithm the statement 'm := m + 1' thus not lead to overflow (i.e., to a value of m exceeding $2^{40} - 1$). For that purpose we need a stronger version of P_0, reading:

$$P_0 = \begin{array}{l} 1 \leq m1/m2 \leq 2 - 2^{-39} \wedge \\ (m1/m2) \times 2^e = g1.m/g2.m \end{array}$$

For proving P_0 we need also a stronger version of invariant I_0:

$$I_0 = \begin{array}{l} (m1 < 2 \times m2 \vee m1 < 2^{40}) \wedge \\ (m1 < 2^{40} \vee even.m1) \wedge \\ (m1/m2) \times 2^e = g1.m/g2.m \end{array}$$

This leads to post condition $I_0 \wedge m1 < 2 \times m2 \wedge m1 \geq m2$, implying:

$$P_0' = \begin{array}{l} 1 \leq m1/m2 < 2 \wedge \\ (m1 < 2^{40} \vee even.m1) \wedge \\ (m1/m2) \times 2^e = g1.m/g2.m \end{array}$$

from which we can infer $m1/m2 \leq 2 - 2^{-39}$ and hence P_0 by the following argument.

We have to find the maximum value of $m1/m2$ under the condition:
$m1/m2 < 2 \wedge m2 < 2^{40} \wedge (m1 < 2^{40} \vee even.m1)$.
Condition $m1/m2 < 2$ implies $m1 \leq 2 \times m2 - 1$. By this restriction $m1/m2$ assumes, for fixed value of $m2$, its maximal value for $m1 = 2 \times m2 - 1$. Then $m1/m2 = 2 - 1/m2$.
For $m2 \leq 2^{39}$ the maximal value of $2 - 1/m2$ is $2 - 2^{-39}$.
For $m2 > 2^{39}$, however, $2 - 1/m2 > 2 - 2^{-39}$. Here we need the additional condition $m1 < 2^{40} \vee even.m1$: $m2 > 2^{39}$ implies $2 \times m2 - 1 > 2^{40}$ and then $m1/m2$ is maximal for an *even* value of $m1 \leq 2 \times m2 - 1$: $m1 = 2 \times m2 - 2$. With this choice of $m1$ we have $m1/m2 = 2 - 2/m2$ and for all values of $m2 \leq 2^{40}$ again $m1/m2 \leq 2 - 2^{-39}$.

We continue the proof by demonstrating:

$\{P_0\}$
m, i := 0, 0;
$\{I_1\}$
do i $<$ 40 $\rightarrow$
if m1 $\geq$ m2 $\rightarrow$ m, m1, i := 2 $*$ m + 1, 2 $*$ (m1 $-$ m2), i + 1
[] m1 $<$ m2 $\rightarrow$ m, m1, i := 2 $*$ m, 2 $*$ m1, i + 1
fi
$\{I_1\}$
od;
$\{P_1\}$

where

$$\begin{aligned} P_1 = \; & 2^{39} \leq m < 2^{40} \wedge \\ & 0 \leq m1/m2 < 2 \wedge \\ & 2^e \leq g1.m/g2.m \leq (2 - 2^{-39}) \times 2^e \wedge \\ & (m + m1/(2 \times m2)) \times 2^{e-39} = g1.m/g2.m \end{aligned}$$

Use for the iteration the following invariant:

$$\begin{aligned} I_1 = \; & 0 \leq i \leq 40 \wedge \\ & 0 \leq m < 2^i \wedge \\ & 0 \leq m1/m2 < 2 \wedge \\ & 2^e \leq g1.m/g2.m \leq (2 - 2^{-39}) \times 2^e \wedge \\ & (m + m1/(2 \times m2)) \times 2^{e-i+1} = g1.m/g2.m \end{aligned}$$

Indeed we have $\{P_0\}$ m, i := 0, 0 $\{I_1\}$, I_1 invariant, and $(I_1 \wedge (i \geq 40)) \Rightarrow P_1$. The fact that in the latter $2^{39} \leq m$ is derived in the following way:

$$\begin{aligned} & 2^{39} \\ \leq \; & (g1.m/g2.m) \times 2^{39-e} \\ = \; & m + m1/(2 \times m2) \\ < \; & m + 1, \end{aligned}$$

therefore $2^{39} \leq m$.

The last step reads:

$\{P_1\}$
if m1 $\geq$ m2 $\rightarrow$ m := m + 1
[] m1 $<$ m2 $\rightarrow$ **skip**
fi;
$\{P_2\}$

where

$$\begin{aligned} P_2 = \; & 2^{39} \leq m < 2^{40} \wedge \\ & (m - 1/2) \times 2^{e-39} \leq g1.m/g2.m < (m + 1/2) \times 2^{e-39} \wedge \\ & (m = 2^{39} \rightarrow m \times 2^{e-39} \leq g1.m/g2.m) \end{aligned}$$

The case $m1 < m2$ is simple: $P_1 \wedge (m1 < m2) \rightarrow P_2$.

The fact that by rounding no overflow can occur, i. e., that the assignment 'm := m + 1' under the condition $m1 \geq m2$ will not lead to $m = 2^{40}$ needs some explanation.

From $P_1 \wedge m1 \geq m2$ we infer:

$$
\begin{array}{ll}
 & m+1 \\
< & m + m1/(2 \times m2) + 1 \\
= & g1.m/g2.m \times 2^{39-e} + 1 \\
\leq & (2 - 2^{-39}) \times 2^e \times 2^{39-e} + 1 \\
= & 2^{40}.
\end{array}
$$

Assertion P_2 implies immediately that $(+1, m, e-39) = rnd.(g1.m/g2.m)$. With s the correct sign of $g1/g2$ it follows that $(s, m, e-39+e1-e2) = rnd.(g1/g2)$.

4.4 Final remarks

If we compare the correctness proof of the division procedure with that of the multiplication procedure, the former seems much simpler than the latter. Nevertheless it was not simple to construct the assertions needed.

First, assertion $m1 < 2 \times m2 \vee m1 < 2^{40}$ in invariant I_0 had to be invented. From an operational point of view it is immediately evident that in the first **do**–loop of the algorithm either $m1$ or $m2$ is scaled up, but never both, in order to arrive at the situation that $1 \leq m1/m2 < 2$. If it is $m2$ that is to be scaled up, i.e. if $m1/m2 \geq 2$, $m1$ remains unaltered and hence $m1 < 2^{40}$. If is $m1$ the one to be scaled up, on the other hand, i.e. if $m1/m2 < 1$, doubling $m1$ will not invalidate $m1/m2 < 2$. It is, however, not clear to me how to find the given assertion by 'letting the formulae doing the work', without any interpretation.

Secondly, assertion $m1/m2 \leq 2 - 2^{-39}$ in P_0, needed to show that rounding of m does not overflow, took me much time to find. Then, for proving that relation in P_0, I had again to add an assertion to I_0. Again that assertion is clear from an operational point of view: initially $m1 < 2^{40}$ and only by doubling it will exceed $2^{40} - 1$, becoming *even* thereby. But again I do not see how to arrive at that assertion for I_0 without that operational interpretation.

Originally I discovered the fact that no overflow can occur when rounding m by the following simple argument.

Let a and b be two natural numbers, $0 \leq a$ and $0 < b < 2^{40}$, such that $a/b < 1$. Then $a < b$, hence $a \leq b - 1$ and therefore $a/b \leq 1 - 1/b < 1 - 2^{-40}$. Rounding a/b to 40 bits precision leads to overflow (i.e., to a result 1) if and only if $a/b + 0.5 \times 2^{-40} \geq 1$, which is not the case.

Now I wonder why I had so much troubles to prove the absence of overflow danger by the technique of weakest preconditions. Did I not use an adequate set of assertions or is that technique not well suited for showing *all* properties of this algorithm correct?

5 Addition

5.1 Operational design considerations

In general the two numbers $f1$ and $f2$ to be added will have different scales, i.e., their binary exponents $f1.e$ and $f2.e$ will be unequal. Before we can add or subtract (depending on whether the two numbers have equal signs or not) the two mantissa's $f1.m$ and $f2.m$ we have to equalize scales. This can be done by decreasing the greatest binary exponent, or by increasing the smallest one, or by doing both.

If we decrease the greatest exponent by one, we have to multiply the corresponding mantissa by two (i.e., shifting it one place to the left). We can do so without capacity problems as long as the latter is less than 2^{39}.

Increasing the smallest exponent by one must be compensated by halving the corresponding mantissa (shifting it one place to the right). Mantissa's being integral, one bit of information is lost.

In order to restrict information loss as much as possible, we have to decrease the greatest exponent as much as possible and only when necessary continue by incrementing the smallest one.

Bits that are lost by shifting the mantissa corresponding to the smallest exponent to the right may play a role in the correct rounding off. We first deal with addition proper, i.e. the case that the two numbers have equal signs.

When adding $f2.m$ to $f1.m$, both mantissa's being at most $2^{40} - 1$, the sum can exceed the capacity. By halving the sum (and incrementing the exponent) we can bring the sum within capacity again, but doing so we loose a bit of information at the same time. In the rounding procedure we need to know whether the fractional part that was shifted out is at least 0.5. It follows that in the case of addition proper we can do with one guarding bit, in which we save the value of the bit that was most recently shifted out (if any at all; otherwise the fractional part is just 0).

Matters are much more complicated in the case of proper subtraction i.e. the case that the two numbers have different signs.

Suppose that $f1.m > f2.m$. We compute $f1.m - f2.m$. Its value is well within the capacity, but, if it is smaller than 2^{39}, we make no good use of the capacity! In that case we should shift it one place to the left again (and, of course, adapt the exponent). Doing so we can reinstall one bit that got lost when equalizing exponents, and we need another bit for the rounding. So we need to keep at least two guarding bits. Do we need more bits?

The answer is both negative and affirmative. Consider the case that, by equalizing the two binary exponents, $f2.m$ is shifted to the right, thereby loosing two or more bits. Then $f1.m \geq 2^{39}$ and $f2.m < 2^{38}$. Therefore $f1.m - f2.m > 2^{38}$ and we can shift this difference at most one place to the left without exceeding the capacity. If we keep the two bits most recently shifted out we always have a rounding bit.

Alas, where in the case of addition proper we round upwards if the fractional part shifted out is at least 0.5, in the case of subtraction proper we round downwards when that

fractional part is more than 0.5. That is the case if the rounding bit is 1 and if at least one of the bits beyond that rounding bit is 1. In stead of keeping all those bits we merely register whether, beyond the two guarding bits, any bit equal to 1 got lost.

For that purpose we keep three guarding bits (hence $0 \leq guard < 8$), of which the least significant one is the bit to register the loss of any non–zero bit beyond the two bits that were shifted out most recently. Consider the case that, by equalizing binary exponents, $f2.m$ is shifted zero or more places to the right, possibly with loss of information. Using $g1$ and $g2$ for the original values (i.e. before shifting) of $f1$ and $f2$, respectively, we have:

$$g2.m \times 2^{g2.e} = (f2.m + \alpha) \times 2^{f2.e}$$

for some rational value α with $0 \leq \alpha < 1$. During shifting out $f2.m$, we keep variable *guard* such that:

case *guard* **of**
$0 : \alpha = 0$
$1 : 0 < \alpha < 1/4$
$2 : \alpha = 1/4$
$3 : 1/4 < \alpha < 1/2$
$4 : \alpha = 1/2$
$5 : 1/2 < \alpha < 3/4$
$6 : \alpha = 3/4$
$7 : 3/4 < \alpha < 1$
end case

For *guard* we have the rule: once odd means odd forever.

For simplicity reasons we compute, during the procedure to equalize the two exponents, the three–bit guard irrespective of the signs of the two numbers (equal or different).

In the case of equal signs we take $m = f1.m + f2.m$ and $e = f1.e$. Then:

$$(m + \alpha) \times 2^e = g1.m \times 2^{g1.e} + g2.m \times 2^{g2.e}$$

If $m \geq 2^{40}$ we have to halve m, to increment e by one, and to adapt *guard* maintaining the above relation.

Now the case of different signs. Let again $f1.m > f2.m$ and take $m = f1.m - f2.m$ and $e = f1.e$. Then:

$$(m - \alpha) \times 2^e = g1.m \times 2^{g1.e} - g2.m \times 2^{g2.e}$$

In order to be able to use a common rounding procedure for both cases (of equal and unequal signs) we rewrite this, for the case that $guard > 0$, as:

$$((m - 1) + (1 - \alpha)) \times 2^e = g1.m \times 2^{g1.e} - g2.m \times 2^{g2.e}$$

and then replace m by $m-1$, $1-\alpha$ by α, and *guard* by $8-guard$. Then we have:

$$(m+\alpha)\times 2^e = g1.m \times 2^{g1.e} - g2.m \times 2^{g2.e}$$

with again the relationship between *guard* and α given above.

If $m < 2^{39}$ we double m and decrement e by 1. This doubles α too, which thereby, in case $guard \geq 4$, become greater than 1. In that case we replace m by $m+1$ (still $m < 2^{40}$, no overflow!) and α by $\alpha - 1$. Of course we have to adapt *guard* too. We do so in the following manner: if $guard < 4$ then replace *guard* by $2 \times guard$ and, otherwise, by $2 \times (guard - 4)$. Once more we have:

$$(m+\alpha)\times 2^e = g1.m \times 2^{g1.e} - g2.m \times 2^{g2.e}$$

now with the following relation between *guard* and α:

case *guard* **of**
$0 : \alpha = 0$
$2 : 0 < \alpha < 1/2$
$4 : \alpha = 1/2$
$6 : 1/2 < \alpha < 1$
end case

We conclude that in all cases we should round upwards if and only if $guard \geq 4$. i.e., $\alpha \geq 1/2$. Again we have to do so carefully: if $m = 2^{40} - 1$ and $guard \geq 4$, the resulting rounded value of m exceeds the mantissa capacity. In that case we set $m = 2^{39}$ and increment e by one.

There are a number of special cases that are dealt with separately. If one of the operands has a zero mantissa, the sum is just the value of the other operand. Also, if the difference of the two exponents is too large, the smaller operand does not contribute to the (rounded) sum. A limiting case where, by rounding, the smaller operand influences the sum is:

$$g1 = (+1, 1, e), g2 = (-1, 2^{39}+1, e-80)$$

Then:

$$g1 + g2 = (2^{40} - 1/2 - 2^{-40}) \times 2^{e-40}$$

rounding to $(2^{40}-1)\times 2^{e-40}$.

If, on the other hand, $g2.e < g1.e - 80$ and $val.g1 \neq 0$, $g2$ is negligible: in that case $rnd.(g1+g2) \cong g1$. For, since $g2.e \leq g1.e - 81$ and $g2.m < 2^{40} \leq g1.m \times 2^{40}$, we have $g2.m \times 2^{g2.e} < g1.m \times 2^{g2.e+40} \leq g1.m \times 2^{g1.e-41}$ or $|val.g2| < |val.g1| \times 2^{-41}$ (cf. Section 2.2).

Note that a sum equal to zero results only if the two operands have equal values but opposite signs:

$$(g1 + g2 = 0) \equiv (val.g1 = -val.g2)$$

5.2 The resulting procedure

```
procedure f_add(f2: f_number);
{computes, for global f1, f1 := f1 + f2}
var m, e, guard: integer; s: sign;
begin {f1 = g1 ∧ f2 = g2}
  let f1, f2 = (s1,m1,e1), (s2,m2,e2);
  if (m1 = 0) or ((e1 < e2 − 80) and (m2 > 0)) → s, m, e := s2, m2, e2
  [] (m2 = 0) or ((e2 < e1 − 80) and (m1 > 0)) → s, m, e := s1, m1, e1
  [] (m1 > 0) and (m2 > 0) and (e1 − 80 ≤ e2 ≤ e1 + 80) →
    guard := 0;
  {equalize binary exponents by shifting the mantissa's:}
    if e1 ≥ e2 →
      do (e1 > e2) and (m1 < 2^39) → m1, e1 := m1 * 2, e1 − 1 od;
      do e1 > e2 →
        if guard in [1,5]        → guard := (m2 mod 2) * 4 + guard div 2 + 1
        [] not (guard in [1,5]) → guard := (m2 mod 2) * 4 + guard div 2
        fi;
        m2, e2 := m2 div 2, e2 + 1
      od
    [] e1 ≤ e2 →
      do (e1 < e2) and (m2 < 2^39) → m2, e2 := m2 * 2, e2 − 1 od;
      do e1 < e2 →
        if guard in [1,5]        → guard := (m1 mod 2) * 4 + guard div 2 + 1
        [] not (guard in [1,5]) → guard := (m1 mod 2) * 4 + guard div 2
        fi;
        m1, e1 := m1 div 2, e1 + 1
      od
    fi;
    e := e1;
  {add or subtract:}
    if s1 = s2 →
      {addition:}
      m, s := m1 + m2, s1;
      if m ≥ 2^40 →
        if guard in [1,5]        → guard := (m mod 2) * 4 + guard div 2 + 1
        [] not (guard in [1,5]) → guard := (m mod 2) * 4 + guard div 2
        fi;
        m, e := m div 2, e + 1
      [] m < 2^40 → skip
      fi;
    [] s1 ≠ s2 →
      {subtraction:}
      if m1 ≥ m2 → m, s := m1 − m2, s1
      [] m1 < m2 → m, s := m2 − m1, s2
      fi;
```

```
        if guard > 0 → m, guard := m − 1, 8 − guard
        [] guard = 0 → skip
        fi;
        if m < 2^39 → m, e, guard := 2 ∗ m + guard div 4, e − 1, (guard mod 4) ∗ 2
        [] m ≥ 2^39 → skip
        fi
      fi;
    {round:}
      if guard ≥ 4 →
        if m = 2^40 − 1 → m, e := 2^39, e + 1
        [] m < 2^40 − 1 → m := m + 1
        fi
      [] guard < 4 → skip
      fi
    fi;
  {form result:}
    f1 := (s,m,e);
    standardize(f1)
  {f1 ≅ rnd.(g1 + g2) ∧ f1 in standard form}
  end {f_add};
```

5.3 Correctness proof of f_add

Once more we have an assertion that holds and is used throughout the proof and which should augment all other assertions:

$$P = \begin{array}{l} 0 \le g1.m < 2^{40} \wedge 0 \le g2.m < 2^{40} \wedge s1 = g1.s \wedge s2 = g2.s \wedge \\ 0 \le m1 < 2^{40} \wedge 0 \le m2 < 2^{40} \wedge 0 \le guard < 8 \end{array}$$

We consider here only the case that both $m1 > 0$ and $m2 > 0$ and $e1 - 80 \le e2 \le e1 + 80$.

We start by proving the correctness of the code for equalizing the two mantissa's. We do so by proving correct a different, but, from an operational view point, equivalent code:

$\{f1 = g1 \wedge f2 = g2 \wedge \mathit{guard} = 0\}$
if e1 $\geq$ e2 $\rightarrow$
  do (e1 > e2) **and** (m1 $< 2^{39}$) $\rightarrow$ m1, e1 := m1 $*$ 2, e1 $-$ 1 **od**;
  $\{P_0\}$
  if e1 > e2 $\rightarrow$
    $\{P_0 \wedge e1 > e2$, hence $I_1\}$
    do e1 > e2 $\rightarrow$
      if *guard* **in** [1,5] $\rightarrow$ guard := (m2 **mod** 2) $*$ 4 + guard **div** 2 + 1
      [] **not** (guard **in** [1,5]) $\rightarrow$ guard := (m2 **mod** 2) $*$ 4 + guard **div** 2
      fi;
      m2, e2 := m2 **div** 2, e2 + 1
      $\{I_1\}$
    od
    $\{P_1\}$
  [] e1 = e2 $\rightarrow$ **skip**
    $\{P_1\}$
  fi
  $\{P_1\}$
[] e1 $\leq$ e2 $\rightarrow \cdots$ {analogous code}
  $\{P_1'\}$
fi
$\{P_1 \vee P_1'\}$

where

$$
\begin{aligned}
P_0 = \;& e1 \geq e2 \wedge \\
& \mathit{guard} = 0 \wedge \\
& (e1 = e2 \vee m1 \geq 2^{39}) \wedge \\
& m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge \\
& m2 \times 2^{e2} = g2.m \times 2^{g2.e}
\end{aligned}
$$

$$
\begin{aligned}
P_1 = \;& e1 = e2 \wedge \\
& (\mathit{guard} = 0 \vee m1 \geq 2^{39}) \wedge \\
& m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge \\
& P_c
\end{aligned}
$$

$$
\begin{aligned}
P_c = \;& \textbf{case } \mathit{guard} \textbf{ of} \\
& 0 : m2 \times 2^{e2} = g2.m \times 2^{g2.e} \\
& 1 : m2 < 2^{37} \wedge m2 \times 2^{e2} < g2.m \times 2^{g2.e} < (m2 + 1/4) \times 2^{e2} \\
& 2 : m2 < 2^{38} \wedge (m2 + 1/4) \times 2^{e2} = g2.m \times 2^{g2.e} \\
& 3 : m2 < 2^{37} \wedge (m2 + 1/4) \times 2^{e2} < g2.m \times 2^{g2.e} < (m2 + 1/2) \times 2^{e2} \\
& 4 : m2 < 2^{39} \wedge (m2 + 1/2) \times 2^{e2} = g2.m \times 2^{g2.e} \\
& 5 : m2 < 2^{37} \wedge (m2 + 1/2) \times 2^{e2} < g2.m \times 2^{g2.e} < (m2 + 3/4) \times 2^{e2} \\
& 6 : m2 < 2^{38} \wedge (m2 + 3/4) \times 2^{e2} = g2.m \times 2^{g2.e} \\
& 7 : m2 < 2^{37} \wedge (m2 + 3/4) \times 2^{e2} < g2.m \times 2^{g2.e} < (m2 + 1) \times 2^{e2} \\
& \textbf{end case}
\end{aligned}
$$

and P_1' and P_c' to be obtained from P_1 and P_c by interchanging $m1$ and $m2$, $e1$ and $e2$, and $g1$ and $g2$, and replacing P_c by P_c'.

The clause $(\mathit{guard} = 0 \vee m1 \geq 2^{39})$ in P_1 guarantees that, in case of loss of precision in

$m2$ ($guard \in 1,3,5,7$), $m1$ was shifted maximally to the left.

For a proof of P_0 use invariant:

$$\begin{aligned} I_0 = \quad & e1 \geq e2 \wedge \\ & guard = 0 \wedge \\ & m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge \\ & m2 \times 2^{e2} = g2.m \times 2^{g2.e} \end{aligned}$$

and for a proof of P_1 invariant I_1, reading:

$$\begin{aligned} I_1 = \quad & e1 \geq e2 \wedge \\ & m1 \geq 2^{39} \wedge \\ & m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge \\ & P_c \end{aligned}$$

Next we first deal with addition proper, i.e. the case that the signs $s1$ and $s2$ are equal. Then we have to add the two mantissa's $m1$ and $m2$ (the two binary exponents being equal now) and to incorporate the guard. We state:

$\{e = e1 = e2 \wedge (P_1 \vee P_1{}') \wedge s1 = s2\}$
m, s := m1 + m2, s1;
$\{P_2\}$
if (m $\geq 2^{40}$) **and** (guard **in** [1,5]) $\rightarrow$
  m, e, guard := m **div** 2, e + 1, (m **mod** 2) $*$ 4 + guard **div** 2 + 1
[] (m $\geq 2^{40}$) **and not** (guard **in** [1,5]) $\rightarrow$
  m, e, guard := m **div** 2, e + 1, (m **mod** 2) $*$ 4 + guard **div** 2
[] m $< 2^{40} \rightarrow$ **skip**
fi
$\{P_3\}$

where

$$\begin{aligned} P_2 = \quad & s = sign.(g1 + g2) \wedge \\ & 0 \leq m < 2^{41} \wedge \\ & (guard = 0 \vee m \geq 2^{39}) \wedge \\ & \textbf{case } guard \textbf{ of} \\ & 0 : m \times 2^e = |g1 + g2| \\ & 1 : m \times 2^e < |g1 + g2| < (m + 1/4) \times 2^e \\ & 2 : (m + 1/4) \times 2^e = |g1 + g2| \\ & 3 : (m + 1/4) \times 2^e < |g1 + g2| < (m + 1/2) \times 2^e \\ & 4 : (m + 1/2) \times 2^e = |g1 + g2| \\ & 5 : (m + 1/2) \times 2^e < |g1 + g2| < (m + 3/4) \times 2^e \\ & 6 : (m + 3/4) \times 2^e = |g1 + g2| \\ & 7 : (m + 3/4) \times 2^e < |g1 + g2| < (m + 1) \times 2^e \\ & \textbf{end case} \end{aligned}$$

and $P_3 = (P_2 \wedge m < 2^{40})$.

If, on the other hand, the two signs $s1$ and $s2$ are unequal, we must subtract the two mantissas. Note that $m1 \geq m2 \wedge guard > 0 \wedge (P_1 \vee P_1{}')$ implies P_1. We state:

$\{e = e1 = e2 \wedge (P_1 \vee P_1{}') \wedge s1 \neq s2\}$
if m1 $\geq$ m2 $\rightarrow$ m, s := m1 $-$ m2, s1
[] m1 $<$ m2 $\rightarrow$ m, s := m2 $-$ m1, s2
fi;
$\{P_4\}$
if guard $> 0 \rightarrow$ m, guard := m $-$ 1, 8 $-$ guard
[] guard $= 0 \rightarrow$ **skip**
fi
$\{P_5\}$

where

$$
\begin{array}{rl}
P_4 = & s = sign.(g1 + g2) \wedge \\
& 0 \leq m < 2^{40} \wedge \\
& \textbf{case } guard \textbf{ of} \\
& 0 : \qquad\qquad\quad m \times 2^e = |g1 + g2| \\
& 1 : m > 2^{38} + 2^{37} \wedge (m - 1/4) \times 2^e < |g1 + g2| < m \times 2^e \\
& 2 : m > 2^{38} \qquad \wedge (m - 1/4) \times 2^e = |g1 + g2| \\
& 3 : m > 2^{38} + 2^{37} \wedge (m - 1/2) \times 2^e < |g1 + g2| < (m - 1/4) \times 2^e \\
& 4 : m > 1 \qquad \wedge (m - 1/2) \times 2^e = |g1 + g2| \\
& 5 : m > 2^{38} + 2^{37} \wedge (m - 3/4) \times 2^e < |g1 + g2| < (m - 1/2) \times 2^e \\
& 6 : m > 2^{38} \qquad \wedge (m - 3/4) \times 2^e = |g1 + g2| \\
& 7 : m > 2^{38} + 2^{37} \wedge (m - 1) \times 2^e < |g1 + g2| < (m - 3/4) \times 2^e \\
& \textbf{end case}
\end{array}
$$

and

$$
\begin{array}{rl}
P_5 = & s = sign.(g1 + g2) \wedge \\
& 0 \leq m < 2^{40} \wedge \\
& \textbf{case } guard \textbf{ of} \\
& 0 : \qquad\qquad\quad m \times 2^e = |g1 + g2| \\
& 1 : m \geq 2^{38} + 2^{37} \wedge \ m \times 2^e < |g1 + g2| < (m + 1/4) \times 2^e \\
& 2 : m \geq 2^{38} \qquad \wedge (m + 1/4) \times 2^e = |g1 + g2 \\
& 3 : m \geq 2^{38} + 2^{37} \wedge (m + 1/4) \times 2^e < |g1 + g2| < (m + 1/2) \times 2^e \\
& 4 : m \geq 1 \qquad \wedge (m + 1/2) \times 2^e = |g1 + g2| \\
& 5 : m \geq 2^{38} + 2^{37} \wedge (m + 1/2) \times 2^e < |g1 + g2| < (m + 3/4) \times 2^e \\
& 6 : m \geq 2^{38} \qquad \wedge (m + 3/4) \times 2^e = |g1 + g2| \\
& 7 : m \geq 2^{38} + 2^{37} \wedge (m + 3/4) \times 2^e < |g1 + g2| < (m + 1) \times 2^e \\
& \textbf{end case}
\end{array}
$$

Next we have:

$\{P_5\}$
if m $< 2^{39} \rightarrow$ m, e, guard := 2 $*$ m + guard **div** 4, e $-$ 1, (guard **mod** 4) $*$ 2 $\{P_6\}$
[] m $\geq 2^{39} \rightarrow$ **skip** $\{P_3\}$
fi
$\{P_3 \vee P_6\}$

with

$$
\begin{aligned}
P_6 = \quad & s = sign.(g1 + g2) \wedge \\
& 0 \leq m < 2^{40} \wedge \\
& (guard = 0 \vee m \geq 2^{39}) \wedge \\
& even.guard \wedge \\
& \textbf{case } guard \textbf{ of} \\
& 0 : m \times 2^e = |g1 + g2| \\
& 2 : m \times 2^e < |g1 + g2| < (m + 1/2) \times 2^e \\
& 4 : (m + 1/2) \times 2^e = |g1 + g2| \\
& 6 : (m + 1/2) \times 2^e < |g1 + g2| < (m + 1) \times 2^e \\
& \textbf{end case}
\end{aligned}
$$

Finally we state:

$\{P_3 \vee P_6\}$
if guard $\geq 4 \rightarrow$
  if m $= 2^{40} - 1 \rightarrow$ m, e $:= 2^{39}$, e + 1
  [] m $< 2^{40} - 1 \rightarrow$ m := m + 1
  fi
[] guard $< 4 \rightarrow$ **skip**
fi
$\{P_7\}$

where

$$
\begin{aligned}
P_7 = \quad & s = sign.(g1 + g2) \wedge \\
& 0 \leq m < 2^{40} \wedge \\
& (m \geq 2^{39} \vee m \times 2^e = |g1 + g2|) \wedge \\
& (m - 1/2) \times 2^e \leq |g1 + g2| < (m + 1/2) \times 2^e \wedge \\
& (m = 2^{39} \rightarrow (m - 1/4) \times 2^e \leq |g1 + g2|)
\end{aligned}
$$

From P_7 we conclude that for $m \geq 2^{39}$ indeed $(s, m, e) = rnd.(g1 + g2)$, whereas for $m < 2^{39}$ we have $(s, |g1 + g2|, 0) \in \mathcal{F} \wedge m \times 2^e = |g1 + g2|$, hence $(s, m, e) \cong rnd.(g1 + g2)$ (c.f. Section 2.2. property 3).

5.4 Final remarks

For the case of addition the correctness proof was lengthy and tedious, but no clever inventions were needed. It was tedious due to the many cases that had to be discriminated, both in the assertions and in the proofs themselves. But it was, having an operational picture in mind, not too hard to device the assertions. In P_c, for example, we have the knowledge that for *odd* values of *guard* at least 3 bits of $m2$ have been shifted out, hence $m2 < 2^{37}$. We use the latter information in P_4 for the conclusion that in these cases $m > 2^{38}$, implying that at most one shift to the left will bring m in the interval $[2^{39}, 2^{40} - 1]$.

Note that P_4 is in fact stronger than is needed for that purpose. The following, slightly weaker version of it would already do the job:

$$
\begin{aligned}
P_4 = \quad & s = \mathit{sign}.(g1 + g2) \wedge \\
& 0 \leq m < 2^{40} \wedge \\
& \textbf{case } \mathit{guard} \textbf{ of} \\
& 0 : \qquad\qquad m \times 2^e = |g1 + g2| \\
& 1 : m > 2^{38} \wedge (m - 1/4) \times 2^e < |g1 + g2| < m \times 2^e \\
& 2 : m > 2^{38} \wedge (m - 1/4) \times 2^e = |g1 + g2| \\
& 3 : m > 2^{38} \wedge (m - 1/2) \times 2^e < |g1 + g2| < (m - 1/4) \times 2^e \\
& 4 : m > 1 \quad \wedge (m - 1/2) \times 2^e = |g1 + g2| \\
& 5 : m > 2^{38} \wedge (m - 3/4) \times 2^e < |g1 + g2| < (m - 1/2) \times 2^e \\
& 6 : m > 2^{38} \wedge (m - 3/4) \times 2^e = |g1 + g2| \\
& 7 : m > 2^{38} \wedge (m - 1) \times 2^e < |g1 + g2| < (m - 3/4) \times 2^e \\
& \textbf{end case}
\end{aligned}
$$

implying also a weaker version of P_5.

The doubling of m in the cases $\mathit{guard} = 0$ or 4 does not necessarily lead to a value of m of at least 2^{39}. That does, however, no harm: in that case the doubled value of m is exact and the rounding step amounts to just a skip.

6 Discussion

The correctness proofs of the three algorithms presented in this report, using the method of weakest–precondition logic, were much harder to conceive and much more tedious than I had anticipated. Certainly if I compare these with the relative ease by which the algorithms themselves were designed using operational arguments.

This can imply that my skillfulness in applying this proof method is too low. But it can also imply that the method is not well suited to this kind of algorithms, and that another proof method would have been more appropriate. I do not really know what is the case here. It would be interesting to see what automated verification could achieve.

Let me try to sum up some of my experiences.

One of the nice things of all the assertions needed in the proofs is that they make explicit, at each stage of the algorithms, what state is arrived at and what knowledge is available. Somehow they embody the operational arguments by which the algorithms were designed.

A first version of the assertions was found just by these operational arguments. In doing the actual correctness proofs some assertions had to be reinforced, and it took some ingenuity to find the reinforcement really needed. Actually, the proofs of the proof obligations were, in general, trivial; a rather restricted number of hints suffices for an easy verification (cf. Appendix A). The work of constructing the correctness proofs was just in the design of the appropriate assertions.

The careful design by operational arguments lead to almost correct algorithms for multiplication, division and addition, and only one small correction of an exceptional case was needed. The correctness proofs were, therefore, essential.

It is often advocated that algorithms should be designed and proved hand in hand, if not the algorithm should be derived from the proofs. But I do not see how, in our case, the latter could have been carried out.

In the actual emulation quite a lot of additional details had to be programmed. The number representation of the EL X8 was in one–complement, with preference for -0 over $+0$ (I have not analyzed whether an implementation directly in the one–complement representation needs more, or perhaps less guarding bits).The binary exponent had to be restricted to 12 bits (including its sign bit), leading to overflow and underflow conditions. The 40 bits of the mantissa had to be divided over two Pascal integer variables. It was a wise decision to use separation of concerns and to *not* incorporate these details in the overall design of the algorithms.

A Proof details

In this appendix we list the proof obligations in terms of the assertions given in the proof sections of the paper. We again use notation

$$Q[x/y]$$

for assertion Q with y substituted for x.

A1: Multiplication

Since neither $g1$ nor $g2$ is changed by the algorithm we take from invariant P only the part:

$0 \leq m1 < 2^{42} \wedge 0 \leq m2$

1. $(P \wedge f1 = g1 \wedge f2 = g2 \wedge g2.m > 0)$
   $\Rightarrow$
   $(P \wedge I_0)\,[m/0, e/0]$

   i.e.

   $(P \wedge f1 = g1 \wedge f2 = g2 \wedge g2.m > 0)$
   $\Rightarrow$
   $(P \wedge$
   $0 \leq 0 \leq m1 \wedge$
   $0 \leq 0 \wedge$
   $(m1 < 2^{41} \rightarrow 0 = 0) \wedge$
   $(m2 > 0 \vee m1 \div 2 \leq 0) \wedge$
   $(0 + m1 \times m2) \times 2^0 \leq g1.m \times g2.m < (0 + 1 + m1 \times m2) \times 2^0$
   $)$

2. $(P \wedge I_0 \wedge m2 > 0 \wedge odd.m2)$
   $\Rightarrow$
   $(P \wedge P_0)\,[m/m + m1, m2/m2 - 1]$

   i.e.

   $(0 \leq m1 < 2^{42} \wedge 0 \leq m2 \wedge$
   $0 \leq m \leq m1 \wedge$
   $0 \leq e \wedge$
   $(m1 < 2^{41} \rightarrow e = 0) \wedge$
   $(m2 > 0 \vee m1 \div 2 \leq m) \wedge$
   $(m + m1 \times m2) \times 2^e \leq g1.m \times g2.m < (m + 1 + m1 \times m2) \times 2^e \wedge$
   $m2 > 0 \wedge$
   $odd.m2$
   $)$
   $\Rightarrow$
   $(0 \leq m1 < 2^{42} \wedge 0 \leq m2 - 1 \wedge$
   $0 \leq m + m1 \leq 2 \times m1 \wedge$
   $0 \leq e \wedge$
   $(m1 < 2^{41} \rightarrow e = 0) \wedge$
   $(m2 - 1 > 0 \vee m1 \leq m + m1) \wedge$
   $(m + m1 + m1 \times (m2 - 1)) \times 2^e \leq g1.m \times g2.m <$
   $\quad (m + m1 + 1 + m1 \times (m2 - 1)) \times 2^e \wedge$
   $even.(m2 - 1)$
   $)$

3. $(P \wedge I_0 \wedge m2 > 0 \wedge even.m2)$
   $\Rightarrow$
   $(P \wedge P_0)$

   i.e.

   $(P \wedge$
   $0 \leq m \leq m1 \wedge$
   $0 \leq e \wedge$
   $(m1 < 2^{41} \rightarrow e = 0) \wedge$
   $(m2 > 0 \vee m1 \div 2 \leq m) \wedge$
   $(m + m1 \times m2) \times 2^e \leq g1.m \times g2.m < (m + 1 + m1 \times m2) \times 2^e \wedge$
   $m2 > 0 \wedge$
   $even.m2$
   $)$
   $\Rightarrow$
   $(P \wedge$
   $0 \leq m \leq 2 \times m1 \wedge$
   $0 \leq e \wedge$
   $(m1 < 2^{41} \rightarrow e = 0) \wedge$
   $(m2 > 0 \vee m1 \leq m) \wedge$
   $(m + m1 \times m2) \times 2^e \leq g1.m \times g2.m < (m + 1 + m1 \times m2) \times 2^e \wedge$
   $even.m2$
   $)$

4. $(P \wedge P_0 \wedge m1 < 2^{41})$
   $\Rightarrow$
   $(P \wedge P_1)\,[m1/2 \times m1]$

   i.e.

   $(0 \leq m1 < 2^{42} \wedge 0 \leq m2 \wedge$
   $0 \leq m \leq 2 \times m1 \wedge$
   $0 \leq e \wedge$
   $(m1 < 2^{41} \rightarrow e = 0) \wedge$
   $(m2 > 0 \vee m1 \leq m) \wedge$
   $(m + m1 \times m2) \times 2^e \leq g1.m \times g2.m < (m + 1 + m1 \times m2) \times 2^e \wedge$
   $even.m2 \wedge$
   $m1 < 2^{41}$
   $)$
   $\Rightarrow$
   $(0 \leq 2 \times m1 < 2^{42} \wedge 0 \leq m2 \wedge$
   $0 \leq m \leq 2 \times m1 \wedge$
   $0 \leq e \wedge$
   $(2 \times m1 < 2^{41} \rightarrow e = 0) \wedge$
   $(m2 > 0 \vee (2 \times m1) \div 2 \leq m) \wedge$
   $(m + 2 \times m1 \times (m2 \div 2)) \times 2^e \leq g1.m \times g2.m < (m + 1 + 2 \times m1 \times (m2 \div 2)) \times 2^e \wedge$
   $even.m2$
   $)$

5. $(P \wedge P_0 \wedge m1 \geq 2^{41})$
   $\Rightarrow$
   $(P \wedge P_1)\,[m/m \div 2, e/e+1]$

   i.e.

$$
\begin{array}{l}
(P \wedge \\
\ 0 \leq m \leq 2 \times m1 \wedge \\
\ 0 \leq e \wedge \\
\ (m1 < 2^{41} \rightarrow e = 0) \wedge \\
\ (m2 > 0 \vee m1 \leq m) \wedge \\
\ (m + m1 \times m2) \times 2^e \leq g1.m \times g2.m < (m + 1 + m1 \times m2) \times 2^e \wedge \\
\ even.m2 \wedge \\
\ m1 \geq 2^{41} \\
) \\
\Rightarrow \\
(P \wedge \\
\ 0 \leq m \div 2 \leq m1 \wedge \\
\ 0 \leq e + 1 \wedge \\
\ (m1 < 2^{41} \rightarrow e + 1 = 0) \wedge \\
\ (m2 > 0 \vee m1 \div 2 \leq m \div 2) \wedge \\
\ (m \div 2 + m1 \times (m2 \div 2)) \times 2^{e+1} \leq g1.m \times g2.m < \\
\quad (m \div 2 + 1 + m1 \times (m2 \div 2)) \times 2^{e+1} \wedge \\
\ even.m2 \wedge \\
)
\end{array}
$$

{Since $m - 1 \leq 2 \times (m \div 2) \leq m$ for integral $m \geq 0$ and $2 \times (m2 \div 2) = m2$ for even $m2$ we have:

$$
\begin{array}{rl}
 & (m \div 2 + m1 \times (m2 \div 2)) \times 2^{e+1} \\
\leq & (m + m1 \times m2) \times 2^e \\
\leq & g1.m \times g2.m \\
< & (m + 1 + m1 \times m2) \times 2^e \\
\leq & (m \div 2 + 1 + m1 \times (m2 \div 2)) \times 2^{e+1}\ \}
\end{array}
$$

6. $(P \wedge P_1)$
   $\Rightarrow$
   $(P \wedge I_0)\,[m2/m2 \div 2]$

   i.e.

   $(0 \leq m1 < 2^{42} \wedge 0 \leq m2 \wedge$
   $0 \leq m \leq m1 \wedge$
   $0 \leq e \wedge$
   $(m1 < 2^{41} \rightarrow e = 0) \wedge$
   $(m2 > 0 \vee m1 \div 2 \leq m) \wedge$
   $(m + m1 \times (m2 \div 2)) \times 2^e \leq g1.m \times g2.m < (m + 1 + m1 \times (m2 \div 2)) \times 2^e \wedge$
   $even.m2$
   $)$
   $\Rightarrow$
   $(0 \leq m1 < 2^{42} \wedge 0 \leq m2 \div 2 \wedge$
   $0 \leq m \leq m1 \wedge$
   $0 \leq e \wedge$
   $(m1 < 2^{41} \rightarrow e = 0) \wedge$
   $(m2 \div 2 > 0 \vee m1 \div 2 \leq m) \wedge$
   $(m + m1 \times (m2 \div 2)) \times 2^e \leq g1.m \times g2.m < (m + 1 + m1 \times (m2 \div 2)) \times 2^e$
   $)$

   {From $even.m2 \wedge m2 > 0$ follows $m2 \geq 2$, hence also $m2 \div 2 > 0$ }

7. $(P \wedge I_0 \wedge m2 = 0)$
   $\Rightarrow$
   $(P \wedge P_2)$

   i.e.

   $(0 \leq m1 < 2^{42} \wedge 0 \leq m2 \wedge$
   $0 \leq m \leq m1 \wedge$
   $0 \leq e \wedge$
   $(m1 < 2^{41} \rightarrow e = 0) \wedge$
   $(m2 > 0 \vee m1 \div 2 \leq m) \wedge$
   $(m + m1 \times m2) \times 2^e \leq g1.m \times g2.m < (m + 1 + m1 \times m2) \times 2^e \wedge$
   $m2 = 0$
   $)$
   $\Rightarrow$
   $(0 \leq m1 < 2^{42} \wedge 0 \leq m2 \wedge$
   $0 \leq m < 2^{42} \wedge$
   $0 \leq e \wedge$
   $(m < 2^{40} \rightarrow e = 0) \wedge$
   $m \times 2^e \leq g1.m \times g2.m < (m + 1) \times 2^e$
   $)$

   $\{ \quad m < 2^{40} \wedge m2 = 0$
   $\Rightarrow \quad m < 2^{40} \wedge m1 \div 2 \leq m$
   $\Rightarrow \quad m1 \div 2 < 2^{40}$
   $\Rightarrow \quad m1 \div 2 \leq 2^{40} - 1$
   $\Rightarrow \quad m1 \leq 2 \times (2^{40} - 1) + 1$
   $\Rightarrow \quad m1 \leq 2^{41} - 1$
   $\Rightarrow \quad m1 < 2^{41}$ $\}$
   $\Rightarrow \quad e = 0$

8. $(P \wedge P_2)$
   $\Rightarrow$
   $(P \wedge I_1)\,[guard = 0]$

   i.e.

   $(P \wedge$
   $0 \le m < 2^{42} \wedge$
   $0 \le e \wedge$
   $(m < 2^{40} \rightarrow e = 0) \wedge$
   $m \times 2^e \le g1.m \times g2.m < (m+1) \times 2^e$
   $)$
   $\Rightarrow$
   $(P \wedge$
   $0 \le m < 2^{42} \wedge$
   $0 \le e \wedge$
   $0 \in [0,1] \wedge$
   $(m < 2^{39} \rightarrow e = 0) \wedge$
   $(e > 0 \vee (0 = 0 \wedge m = g1.m \times g2.m)) \wedge$
   $(m + 0/2) \times 2^e \le g1.m \times g2.m < (m+1) \times 2^e \wedge$
   $(m < 2^{40} \rightarrow g1.m \times g2.m < (m + 0/2 + 1/2) \times 2^e)$
   $)$

9. $(P \wedge I_1 \wedge m \ge 2^{40})$
   $\Rightarrow$
   $(P \wedge I_1)\,[guard/m \bmod 2, m/m \div 2, e/e+1]$

   i.e.

   $(P \wedge$
   $0 \le m < 2^{42} \wedge$
   $0 \le e \wedge$
   $guard \in [0,1] \wedge$
   $(m < 2^{39} \rightarrow e = 0) \wedge$
   $(e > 0 \vee (guard = 0 \wedge m = g1.m \times g2.m)) \wedge$
   $(m + guard/2) \times 2^e \le g1.m \times g2.m < (m+1) \times 2^e \wedge$
   $(m < 2^{40} \rightarrow g1.m \times g2.m < (m + guard/2 + 1/2) \times 2^e) \wedge$
   $m \ge 2^{40}$
   $)$
   $\Rightarrow$
   $(P \wedge$
   $0 \le m \div 2 < 2^{42} \wedge$
   $0 \le e + 1 \wedge$
   $m \bmod 2 \in [0,1] \wedge$
   $(m \div 2 < 2^{39} \rightarrow e + 1 = 0) \wedge$
   $(e + 1 > 0 \vee (m \bmod 2 = 0 \wedge m \div 2 = g1.m \times g2.m)) \wedge$
   $(m \div 2 + (m \bmod 2)/2) \times 2^{e+1} \le g1.m \times g2.m < (m \div 2 + 1) \times 2^{e+1} \wedge$
   $(m \div 2 < 2^{40} \rightarrow g1.m \times g2.m < (m \div 2 + (m \bmod 2)/2 + 1/2) \times 2^{e+1})$
   $)$
   $\{m = 2 \times (m \div 2 + (m \bmod 2)/2)$, hence $m + 1 \le 2 \times (m \div 2) + 2$ $\}$

10. $(P \wedge I_1 \wedge m < 2^{40})$
    $\Rightarrow$
    $(P \wedge P_3)$

    i.e.

    $(P \wedge$
    $0 \leq m < 2^{42} \wedge$
    $0 \leq e \wedge$
    $guard \in [0,1] \wedge$
    $(m < 2^{39} \rightarrow e = 0) \wedge$
    $(e > 0 \vee (guard = 0 \wedge m = g1.m \times g2.m)) \wedge$
    $(m + guard/2) \times 2^e \leq g1.m \times g2.m < (m+1) \times 2^e \wedge$
    $(m < 2^{40} \rightarrow g1.m \times g2.m < (m + guard/2 + 1/2) \times 2^e) \wedge$
    $m < 2^{40}$
    $)$
    $\Rightarrow$
    $(P \wedge$
    $0 \leq m < 2^{40} \wedge$
    $0 \leq e \wedge$
    $guard \in [0,1] \wedge$
    $(m < 2^{39} \rightarrow e = 0) \wedge$
    $(e > 0 \vee (guard = 0 \wedge m = g1.m \times g2.m)) \wedge$
    $(m + guard/2) \times 2^e \leq g1.m \times g2.m < (m + guard/2 + 1/2) \times 2^e$
    $)$

11. $(P \wedge P3 \wedge guard = 1 \wedge m = 2^{40} - 1)$
    $\Rightarrow$
    $(P \wedge R)\,[m/2^{39}, e/e+1]$

    i.e.

    $(P \wedge$
    $0 \leq m < 2^{40} \wedge$
    $0 \leq e \wedge$
    $guard \in [0,1] \wedge$
    $(m < 2^{39} \rightarrow e = 0) \wedge$
    $(e > 0 \vee (guard = 0 \wedge m = g1.m \times g2.m)) \wedge$
    $(m + guard/2) \times 2^e \leq g1.m \times g2.m < (m + guard/2 + 1/2) \times 2^e \wedge$
    $guard = 1 \wedge$
    $m = 2^{40} - 1$
    $)$
    $\Rightarrow$
    $(P \wedge$
    $0 \leq 2^{39} < 2^{40} \wedge$
    $0 \leq e + 1 \wedge$
    $(2^{39} < 2^{39} \rightarrow e + 1 = 0) \wedge$
    $(e + 1 > 0 \vee 2^{39} = g1.m \times g2.m) \wedge$
    $(2^{39} - 1/2) \times 2^{e+1} \leq g1.m \times g2.m < (2^{39} + 1/2) \times 2^{e+1} \wedge$
    $(2^{39} = 2^{39} \rightarrow (2^{39} - 1/4) \times 2^{e+1} \leq g1.m \times g2.m)$
    $)$

12. $(P \wedge P3 \wedge guard = 1 \wedge m < 2^{40} - 1)$
$\Rightarrow$
$(P \wedge R)\,[m/m+1]$

i.e.

$(P \wedge$
$0 \leq m < 2^{40} \wedge$
$0 \leq e \wedge$
$guard \in [0,1] \wedge$
$(m < 2^{39} \rightarrow e = 0) \wedge$
$(e > 0 \vee (guard = 0 \wedge m = g1.m \times g2.m)) \wedge$
$(m + guard/2) \times 2^e \leq g1.m \times g2.m < (m + guard/2 + 1/2) \times 2^e \wedge$
$guard = 1 \wedge$
$m < 2^{40} - 1$
$)$
$\Rightarrow$
$(P \wedge$
$0 \leq m + 1 < 2^{40} \wedge$
$0 \leq e \wedge$
$(m + 1 < 2^{39} \rightarrow e = 0) \wedge$
$(e > 0 \vee m + 1 = g1.m \times g2.m) \wedge$
$(m + 1 - 1/2) \times 2^e \leq g1.m \times g2.m < (m + 1 + 1/2) \times 2^e \wedge$
$(m + 1 = 2^{39} \rightarrow (m + 1 - 1/4) \times 2^e \leq g1.m \times g2.m)$
$)$

$\{guard = 1$ implies $e > 0$, hence $m \geq 2^{39}$ and $\neg\,(m + 1 = 2^{39})\ \}$

13. $(P \wedge P3 \wedge guard = 0)$
$\Rightarrow$
$(P \wedge R)$

i.e.

$(P \wedge$
$0 \leq m < 2^{40} \wedge$
$0 \leq e \wedge$
$guard \in [0,1] \wedge$
$(m < 2^{39} \rightarrow e = 0) \wedge$
$(e > 0 \vee (guard = 0 \wedge m = g1.m \times g2.m)) \wedge$
$(m + guard/2) \times 2^e \leq g1.m \times g2.m < (m + guard/2 + 1/2) \times 2^e \wedge$
$guard = 0$
$)$
$\Rightarrow$
$(P \wedge$
$0 \leq m < 2^{40} \wedge$
$0 \leq e \wedge$
$(m < 2^{39} \rightarrow e = 0) \wedge$
$(e > 0 \vee m = g1.m \times g2.m) \wedge$
$(m - 1/2) \times 2^e \leq g1.m \times g2.m < (m + 1/2) \times 2^e \wedge$
$(m = 2^{39} \rightarrow (m - 1/4) \times 2^e \leq g1.m \times g2.m)$
$)$

A2: Division

Since neither $g1$ nor $g2$ is changed by the algorithm we take for invariant P only the part:

$0 \leq m1 < 2^{41} \wedge 1 \leq m2 < 2^{40}$

(Recall that we consider only the case that both $g1.m > 0$ and $g2.m > 0$.)

1. $(P \wedge f1 = g1 \wedge f2 = g2)$
   $\Rightarrow$
   $(P \wedge I_0)\,[e/0]$

   i.e.

   $(P \wedge f1 = g1 \wedge f2 = g2)$
   $\Rightarrow$
   $(P \wedge$
   $(m1 < 2 \times m2 \vee m1 < 2^{40}) \wedge$
   $(m1 < 2^{40} \vee even.m1) \wedge$
   $(m1/m2) \times 2^0 = g1.m/g2.m$
   $)$

   $\{g1.m < 2^{40}$, hence $m1 < 2^{40}\}$

2. $(P \wedge I_0 \wedge m1 \geq 2 \times m2)$
   $\Rightarrow$
   $(P \wedge I_0)\,[m2/2 \times m2, e/e + 1]$

   i.e.

   $(0 \leq m1 < 2^{41} \wedge 1 \leq m2 < 2^{40} \wedge$
   $(m1 < 2 \times m2 \vee m1 < 2^{40}) \wedge$
   $(m1 < 2^{40} \vee even.m1) \wedge$
   $(m1/m2) \times 2^e = g1.m/g2.m \wedge$
   $m1 \geq 2 \times m2$
   $)$
   $\Rightarrow$
   $(0 \leq m1 < 2^{41} \wedge 1 \leq 2 \times m2 < 2^{40} \wedge$
   $(m1 < 2 \times 2 \times m2 \vee m1 < 2^{40}) \wedge$
   $(m1 < 2^{40} \vee even.m1) \wedge$
   $(m1/(2 \times m2) \times 2^{e+1} = g1.m/g2.m$
   $)$

   $\{m1 \geq 2 \times m2$, hence $m1 < 2^{40}$ and so $2 \times m2 < 2^{40}\}$

3. $(P \wedge I_0 \wedge m1 < m2)$
   $\Rightarrow$
   $(P \wedge I_0)\,[m1/2 \times m1, e/e-1]$

   i.e.

   $(0 \leq m1 < 2^{41} \wedge 1 \leq m2 < 2^{40} \wedge$
   $(m1 < 2 \times m2 \vee m1 < 2^{40}) \wedge$
   $(m1 < 2^{40} \vee even.m1) \wedge$
   $(m1/m2) \times 2^e = g1.m/g2.m \wedge$
   $m1 < m2$
   $)$
   $\Rightarrow$
   $(0 \leq 2 \times m1 < 2^{41} \wedge 1 \leq m2 < 2^{40} \wedge$
   $(2 \times m1 < 2 \times m2 \vee 2 \times m1 < 2^{40}) \wedge$
   $(2 \times m1 < 2^{40} \vee even.(2 \times m1)) \wedge$
   $((2 \times m1)/m2) \times 2^{e-1} = g1.m/g2.m$
   $)$

   $\{m1 < m2$, hence $m1 < 2^{40}\}$

4. $(P \wedge I_0 \wedge m1 < 2 \times m2 \wedge m1 \geq m2)$
   $\Rightarrow$
   $(P \wedge P_0{}')$

   i.e.

   $(P \wedge$
   $(m1 < 2 \times m2 \vee m1 < 2^{40}) \wedge$
   $(m1 < 2^{40} \vee even.m1) \wedge$
   $(m1/m2) \times 2^e = g1.m/g2.m \wedge$
   $m1 < 2 \times m2 \wedge$
   $m1 \geq m2$
   $)$
   $\Rightarrow$
   $(P \wedge$
   $1 \leq m1/m2 < 2 \wedge$
   $(m1 < 2^{40} \vee even.m1) \wedge$
   $(m1/m2) \times 2^e = g1.m/g2.m$
   $)$

5. $(P \wedge P_0)$
   $\Rightarrow$
   $(P \wedge I_1)\,[m/0, i/0]$

   i.e.

   $(P \wedge$
   $1 \leq m1/m2 \leq 2 - 2^{-39} \wedge$
   $(m1/m2) \times 2^e = g1.m/g2.m$
   $)$
   $\Rightarrow$
   $(P \wedge$
   $0 \leq 0 \leq 40 \wedge$
   $0 \leq 0 < 2^0 \wedge$
   $0 \leq m1/m2 < 2 \wedge$
   $2^e \leq g1.m/g2.m \leq (2 - 2^{-39}) \times 2^e \wedge$
   $(0 + m1/(2 \times m2)) \times 2^{e-0+1} = g1.m/g2.m$
   $)$

6. $(P \wedge I_1 \wedge i < 40 \wedge m1 \geq m2)$
   $\Rightarrow$
   $(P \wedge I_1)\,[m/2 \times m + 1, m1/2 \times (m1 - m2), i/i + 1]$

   i.e.

   $(0 \leq m1 < 2^{41} \wedge 1 \leq m2 < 2^{40} \wedge$
   $0 \leq i \leq 40 \wedge$
   $0 \leq m < 2^i \wedge$
   $0 \leq m1/m2 < 2 \wedge$
   $2^e \leq g1.m/g2.m \leq (2 - 2^{-39}) \times 2^e \wedge$
   $(m + m1/(2 \times m2)) \times 2^{e-i+1} = g1.m/g2.m \wedge$
   $i < 40 \wedge$
   $m1 \geq m2$
   $)$
   $\Rightarrow$
   $(0 \leq 2 \times (m1 - m2) < 2^{41} \wedge 1 \leq m2 < 2^{40} \wedge$
   $0 \leq i + 1 \leq 40 \wedge$
   $0 \leq 2 \times m + 1 < 2^{i+1} \wedge$
   $0 \leq (2 \times (m1 - m2))/m2 < 2 \wedge$
   $2^e \leq g1.m/g2.m \leq (2 - 2^{-39}) \times 2^e \wedge$
   $((2 \times m + 1 + 2 \times (m1 - m2))/(2 \times m2)) \times 2^{e-(i+1)+1} = g1.m/g2.m$
   $)$

   $\{m2/m1 > 1/2$, hence
   $\begin{array}{rl} & 2 \times (m1 - m2) \\ = & 2 \times m1 \times (1 - m2/m1) \\ < & 2 \times m1 \times (1 - 1/2) \\ = & m1\}\end{array}$

7. $(P \wedge I_1 \wedge i < 40 \wedge m1 < m2)$
   $\Rightarrow$
   $(P \wedge I_1)\,[m/2 \times m, m1/2 \times m1, i/i+1]$

   i.e.

   $(0 \leq m1 < 2^{41} \wedge 1 \leq m2 < 2^{40} \wedge$
   $0 \leq i \leq 40 \wedge$
   $0 \leq m < 2^i \wedge$
   $0 \leq m1/m2 < 2 \wedge$
   $2^e \leq g1.m/g2.m \leq (2 - 2^{-39}) \times 2^e \wedge$
   $(m + m1/(2 \times m2)) \times 2^{e-i+1} = g1.m/g2.m \wedge$
   $i < 40 \wedge$
   $m1 < m2$
   $)$
   $\Rightarrow$
   $(0 \leq 2 \times m1 < 2^{41} \wedge 1 \leq m2 < 2^{40} \wedge$
   $0 \leq i + 1 \leq 40 \wedge$
   $0 \leq 2 \times m < 2^{i+1} \wedge$
   $0 \leq (2 \times m1)/m2 < 2 \wedge$
   $2^e \leq g1.m/g2.m \leq (2 - 2^{-39}) \times 2^e \wedge$
   $(2 \times m + (2 \times m1)/(2 \times m2)) \times 2^{e-(i+1)+1} = g1.m/g2.m$
   $)$

   $\{m1 < m2$, hence
   $\begin{array}{rl} & 2 \times m1 \\ < & 2 \times m2 \\ < & 2 \times 2^{40} \\ = & 2^{41}\}\end{array}$

8. $(P \wedge I_1 \wedge i \geq 40)$
   $\Rightarrow$
   $(P \wedge P_1)$

   i.e.

   $(P \wedge$
   $0 \leq i \leq 40 \wedge$
   $0 \leq m < 2^i \wedge$
   $0 \leq m1/m2 < 2 \wedge$
   $2^e \leq g1.m/g2.m \leq (2 - 2^{-39}) \times 2^e \wedge$
   $(m + m1/(2 \times m2)) \times 2^{e-i+1} = g1.m/g2.m \wedge$
   $i \geq 40$
   $)$
   $\Rightarrow$
   $(P \wedge$
   $2^{39} \leq m < 2^{40} \wedge$
   $0 \leq m1/m2 < 2 \wedge$
   $2^e \leq g1.m/g2.m \leq (2 - 2^{-39}) \times 2^e \wedge$
   $(m + m1/(2 \times m2)) \times 2^{e-39} = g1.m/g2.m$
   $)$

   $\{\begin{array}{rl} & 2^{39} \\ \leq & (g1.m/g2.m) \times 2^{39-e} \\ = & m + m1/(2 \times m2) \\ < & m + 1, \end{array}$
   hence $2^{39} \leq m\}$

9. $(P \wedge P_1 \wedge m1 \geq m2)$
   $\Rightarrow$
   $(P \wedge P_2)\,[m/m+1]$

   i.e.

   $$
   \begin{array}{l}
   (P \wedge \\
   \quad 2^{39} \leq m < 2^{40} \wedge \\
   \quad 0 \leq m1/m2 < 2 \wedge \\
   \quad 2^e \leq g1.m/g2.m \leq (2 - 2^{-39}) \times 2^e \wedge \\
   \quad (m + m1/(2 \times m2)) \times 2^{e-39} = g1.m/g2.m \wedge \\
   \quad m1 \geq m2 \\
) \\
   \Rightarrow \\
   (P \wedge \\
   \quad 2^{39} \leq m + 1 < 2^{40} \wedge \\
   \quad (m + 1 - 1/2) \times 2^{e-39} \leq g1.m/g2.m < (m + 1 + 1/2) \times 2^{e-39} \wedge \\
   \quad (m + 1 = 2^{39} \rightarrow (m + 1) \times 2^{e-39} \leq g1.m/g2.m) \\
)
   \end{array}
   $$

   $$
   \begin{array}{ll}
   \{ & m + 1 \\
   < & m + m1/(2 \times m2) + 1 \\
   = & (g1.m/g2.m) \times 2^{39-e} + 1 \\
   \leq & (2 - 2^{-39}) \times 2^{39} + 1 \\
   = & 2^{40}\}
   \end{array}
   $$

10. $(P \wedge P_1 \wedge m1 < m2)$
    $\Rightarrow$
    $(P \wedge P_2)$

    i.e.

    $$
    \begin{array}{l}
    (P \wedge \\
    \quad 2^{39} \leq m < 2^{40} \wedge \\
    \quad 0 \leq m1/m2 < 2 \wedge \\
    \quad 2^e \leq g1.m/g2.m \leq (2 - 2^{-39}) \times 2^e \wedge \\
    \quad (m + m1/(2 \times m2)) \times 2^{e-39} = g1.m/g2.m \wedge \\
    \quad m1 < m2 \\
) \\
    \Rightarrow \\
    (P \wedge \\
    \quad 2^{39} \leq m < 2^{40} \wedge \\
    \quad (m - 1/2) \times 2^{e-39} \leq g1.m/g2.m < (m + 1/2) \times 2^{e-39} \wedge \\
    \quad (m = 2^{39} \rightarrow m \times 2^{e-39} \leq g1.m/g2.m) \\
)
    \end{array}
    $$

A3: Addition

Since $g1$, $g2$, $s1$, nor $s2$ are changed by the algorithm we take from invariant P only the part:

$0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq guard < 8$

1. $(P \wedge f1 = g1 \wedge f2 = g2 \wedge guard = 0 \wedge e1 \geq e2)$
   $\Rightarrow$
   $(P \wedge I_0)$

   i.e.

   $(P \wedge f1 = g1 \wedge f2 = g2 \wedge guard = 0 \wedge e1 \geq e2)$
   $\Rightarrow$
   $(P \wedge$
   $e1 \geq e2 \wedge$
   $guard = 0 \wedge$
   $m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
   $m2 \times 2^{e2} = g2.m \times 2^{g2.e}$
   $)$

2. $(P \wedge I_0 \wedge e1 > e2 \wedge m1 < 2^{39})$
   $\Rightarrow$
   $(P \wedge I_0)\,[m1/m1 \times 2, e1/e1 - 1]$

   i.e.

   $(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq guard < 8 \wedge$
   $e1 \geq e2 \wedge$
   $guard = 0 \wedge$
   $m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
   $m2 \times 2^{e2} = g2.m \times 2^{g2.e} \wedge$
   $e1 > e2 \wedge$
   $m1 < 2^{39}$
   $)$
   $\Rightarrow$
   $(0 \leq m1 \times 2 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq guard < 8 \wedge$
   $e1 - 1 \geq e2 \wedge$
   $guard = 0 \wedge$
   $m1 \times 2 \times 2^{e1-1} = g1.m \times 2^{g1.e} \wedge$
   $m2 \times 2^{e2} = g2.m \times 2^{g2.e}$
   $)$

3. $(P \wedge I_0 \wedge (e1 \leq e2 \vee m1 \geq 2^{39}))$
   $\Rightarrow$
   $(P \wedge P_0)$

   i.e.

   $(P \wedge$
   $e1 \geq e2 \wedge$
   $guard = 0 \wedge$
   $m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
   $m2 \times 2^{e2} = g2.m \times 2^{g2.e} \wedge$
   $(e1 \leq e2 \vee m1 \geq 2^{39})$
   $)$
   $\Rightarrow$
   $(P \wedge$
   $e1 \geq e2 \wedge$
   $guard = 0 \wedge$
   $(e1 = e2 \vee m1 \geq 2^{39}) \wedge$
   $m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
   $m2 \times 2^{e2} = g2.m \times 2^{g2.e}$
   $)$

4. $(P \wedge P_0 \wedge e1 > e2)$
   $\Rightarrow$
   $(P \wedge I_1)$

   i.e.

   $(P \wedge$
   $e1 \geq e2 \wedge$
   $guard = 0 \wedge$
   $(e1 = e2 \vee m1 \geq 2^{39}) \wedge$
   $m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
   $m2 \times 2^{e2} = g2.m \times 2^{g2.e} \wedge$
   $e1 > e2$
   $)$
   $\Rightarrow$
   $(P \wedge$
   $e1 \geq e2 \wedge$
   $m1 \geq 2^{39} \wedge$
   $m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
   $guard = 0 \wedge m2 \times 2^{e2} = g2.m \times 2^{g2.e}$
   $)$

5. {two out of 16 cases[4]:}

5a. $(P \wedge I_1 \wedge e1 > e2 \wedge guard = 3 \wedge m2 = 2 \times (m2 \div 2))$
$\Rightarrow$
$(P \wedge I_1)\,[guard/(m2 \bmod 2) \times 4 + guard \div 2, m2/m2 \div 2, e2/e2 + 1]$
i.e.
$(0 \le m1 < 2^{40} \wedge 0 \le m2 < 2^{40} \wedge 0 \le guard < 8 \wedge$
$e1 \ge e2 \wedge$
$m1 \ge 2^{39} \wedge$
$m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
$guard = 3 \wedge m2 < 2^{37} \wedge (m2 + 1/4) \times 2^{e2} < g2.m \times 2^{g2.e} < (m2 + 1/2) \times 2^{e2} \wedge$
$e1 > e2 \wedge$
$m2 = 2 \times (m2 \div 2)$
$)$
$\Rightarrow$
$(0 \le m1 < 2^{40} \wedge 0 \le m2 \div 2 < 2^{40} \wedge 0 \le (m2 \bmod 2) \times 4 + guard \div 2 < 8 \wedge$
$e1 \ge e2 + 1 \wedge$
$m1 \ge 2^{39} \wedge$
$m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
$(m2 \bmod 2) \times 4 + guard \div 2 = 1 \wedge m2 \div 2 < 2^{37} \wedge$
$m2 \div 2 \times 2^{e2+1} < g2.m \times 2^{g2.e} < (m2 \div 2 + 1/4) \times 2^{e2+1}$
$)$[5]

5b. $(P \wedge I_1 \wedge e1 > e2 \wedge guard = 4 \wedge m2 = 2 \times (m2 \div 2) + 1)$
$\Rightarrow$
$(P \wedge I_1)\,[guard/(m2 \bmod 2) \times 4 + guard \div 2, m2/m2 \div 2, e2/e2 + 1]$
i.e.
$(0 \le m1 < 2^{40} \wedge 0 \le m2 < 2^{40} \wedge 0 \le guard < 8 \wedge$
$e1 \ge e2 \wedge$
$m1 \ge 2^{39} \wedge$
$m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
$guard = 4 \wedge m2 < 2^{39} \wedge (m2 + 1/2) \times 2^{e2} = g2.m \times 2^{g2.e} \wedge$
$e1 > e2 \wedge$
$m2 = 2 \times (m2 \div 2) + 1$
$)$
$\Rightarrow$
$(0 \le m1 < 2^{40} \wedge 0 \le m2 \div 2 < 2^{40} \wedge 0 \le (m2 \bmod 2) \times 4 + guard \div 2 < 8 \wedge$
$e1 \ge e2 + 1 \wedge$
$m1 \ge 2^{39} \wedge$
$m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
$(m2 \bmod 2) \times 4 + guard \div 2 = 6 \wedge m2 \div 2 < 2^{38} \wedge$
$(m2 \div 2 + 3/4) \times 2^{e2+1} = g2.m \times 2^{g2.e}$
$)$

[4] 16 cases by eight values for *guard* combined with odd or even values of $m2$.
[5] leading, after the assignment, to P_c with $guard = 1$.

6. $(P \wedge I_1 \wedge e1 \leq e2)$
   $\Rightarrow$
   $(P \wedge P_1)$

   i.e.

   $(P \wedge$
   $e1 \geq e2 \wedge$
   $m1 \geq 2^{39} \wedge$
   $m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
   $P_c \wedge$
   $e1 \leq e2$
   $)$
   $\Rightarrow$
   $(P \wedge$
   $e1 = e2 \wedge$
   $(guard = 0 \vee m1 \geq 2^{39}) \wedge$
   $m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
   P_c
   $)$

7. $(P \wedge P_0 \wedge e1 = e2)$
   $\Rightarrow$
   $(P \wedge P_1)$

   i.e.

   $(P \wedge$
   $e1 \geq e2 \wedge$
   $guard = 0 \wedge$
   $(e1 = e2 \vee m1 \geq 2^{39}) \wedge$
   $m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
   $m2 \times 2^{e2} = g2.m \times 2^{g2.e} \wedge$
   $e1 = e2$
   $)$
   $\Rightarrow$
   $(P \wedge$
   $e1 = e2 \wedge$
   $(guard = 0 \vee m1 \geq 2^{39}) \wedge$
   $m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
   $guard = 0 \wedge m2 \times 2^{e2} = g2.m \times 2^{g2.e}$
   $)$

8. {one out of 8 cases:}

$(P \wedge e = e1 = e2 \wedge (P_1 \vee P_1{}') \wedge guard = 7 \wedge s1 = s2)$
$\Rightarrow$
$(P \wedge P_2)\,[m/m1 + m2, s/s1]$

i.e.

$(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq guard < 8 \wedge$
$e = e1 = e2 \wedge$
$(e1 = e2 \wedge$
$(guard = 0 \vee m1 \geq 2^{39}) \wedge$
$m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
$guard = 7 \wedge m2 < 2^{37} \wedge (m2 + 3/4) \times 2^{e2} < g2.m \times 2^{g2.e} < (m2 + 1) \times 2^{e2}$
$\vee$
$e2 = e1 \wedge$
$(guard = 0 \vee m2 \geq 2^{39}) \wedge$
$m2 \times 2^{e2} = g2.m \times 2^{g2.e} \wedge$
$guard = 7 \wedge m1 < 2^{37} \wedge (m1 + 3/4) \times 2^{e1} < g1.m \times 2^{g1.e} < (m1 + 1) \times 2^{e1}$
$) \wedge$
$s1 = s2$
$)$
$\Rightarrow$
$(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq guard < 8 \wedge$
$s1 = sign.(g1 + g2) \wedge$
$0 \leq m1 + m2 < 2^{41} \wedge$
$(guard = 0 \vee m1 + m2 \geq 2^{39}) \wedge$
$guard = 7 \wedge (m1 + m2 + 3/4) \times 2^{e} < |g1 + g2| < (m1 + m2 + 1) \times 2^{e}$
$)$

9. {two out of 16 cases}:

9a: $(P \wedge P_2 \wedge m \geq 2^{40} \wedge \mathit{guard} = 5 \wedge m = 2 \times (m \div 2))$
$\Rightarrow$
$(P \wedge P_3)\,[m/m \div 2, e/e + 1, \mathit{guard}/(m \bmod 2) \times 4 + \mathit{guard} \div 2 + 1]$

i.e.

$(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq \mathit{guard} < 8 \wedge$
$s = \mathit{sign}.(g1 + g2) \wedge$
$0 \leq m < 2^{41} \wedge$
$(\mathit{guard} = 0 \vee m \geq 2^{39}) \wedge$
$\mathit{guard} = 5 \wedge (m + 1/2) \times 2^e < |g1 + g2| < (m + 3/4) \times 2^e \wedge$
$m \geq 2^{40} \wedge$
$m = 2 \times (m \div 2)$
$)$
$\Rightarrow$
$(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq (m \bmod 2) \times 4 + \mathit{guard} \div 2 + 1 < 8 \wedge$
$s = \mathit{sign}.(g1 + g2) \wedge$
$0 \leq m \div 2 < 2^{41} \wedge$
$((m \bmod 2) \times 4 + \mathit{guard} \div 2 + 1 = 0 \vee m \div 2 \geq 2^{39}) \wedge$
$(m \bmod 2) \times 4 + \mathit{guard} \div 2 + 1 = 3 \wedge$
$\quad (m \div 2 + 1/4) \times 2^{e+1} < |g1 + g2| < (m \div 2 + 1/2) \times 2^{e+1}$
$m \div 2 < 2^{40}$
$)$

9b: $(P \wedge P_2 \wedge m \geq 2^{40} \wedge \mathit{guard} = 4 \wedge m = 2 \times (m \div 2) + 1)$
$\Rightarrow$
$(P \wedge P_3)\,[m/m \div 2, e/e + 1, \mathit{guard}/(m \bmod 2) \times 4 + \mathit{guard} \div 2]$

i.e.

$(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq \mathit{guard} < 8 \wedge$
$s = \mathit{sign}.(g1 + g2) \wedge$
$0 \leq m < 2^{41} \wedge$
$(\mathit{guard} = 0 \vee m \geq 2^{39}) \wedge$
$\mathit{guard} = 4 \wedge (m + 1/2) \times 2^e = |g1 + g2| \wedge$
$m \geq 2^{40} \wedge$
$m = 2 \times (m \div 2) + 1$
$)$
$\Rightarrow$
$(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq (m \bmod 2) \times 4 + \mathit{guard} \div 2 < 8 \wedge$
$s = \mathit{sign}.(g1 + g2) \wedge$
$0 \leq m \div 2 < 2^{41} \wedge$
$((m \bmod 2) \times 4 + \mathit{guard} \div 2 = 0 \vee m \div 2 \geq 2^{39}) \wedge$
$(m \bmod 2) \times 4 + \mathit{guard} \div 2 = 6 \wedge (m \div 2 + 3/4) \times 2^{e+1} = |g1 + g2|$
$m \div 2 < 2^{40}$
$)$

10. $(P \wedge P_2 \wedge m < 2^{40})$
    $\Rightarrow$
    $(P \wedge P_3)$

    i.e.

    $(P \wedge P_2 \wedge m < 2^{40})$
    $\Rightarrow$
    $(P \wedge P_2 \wedge m < 2^{40})$

11. one out of 8 cases

    $(P \wedge e = e1 = e2 \wedge (P_1 \vee P_1{}') \wedge guard = 1 \wedge s1 \neq s2 \wedge m1 \geq m2)$
    $\Rightarrow$
    $(P \wedge P_4)\,[m/m1 - m2, s/s1]$

    i.e.

    $(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq guard < 8 \wedge$
    $e = e1 = e2 \wedge$
    $(e1 = e2 \wedge$
    $(guard = 0 \vee m1 \geq 2^{39}) \wedge$
    $m1 \times 2^{e1} = g1.m \times 2^{g1.e} \wedge$
    $guard = 1 \wedge m2 < 2^{37} \wedge m2 \times 2^{e2} < g2.m \times 2^{g2.e} < (m2 + 1/4) \times 2^{e2}$
    $\vee$
    $e2 = e1 \wedge$
    $(guard = 0 \vee m2 \geq 2^{39}) \wedge$
    $m2 \times 2^{e2} = g2.m \times 2^{g2.e} \wedge$
    $guard = 1 \wedge m1 < 2^{37} \wedge m1 \times 2^{e1} < g1.m \times 2^{g1.e} < (m1 + 1/4) \times 2^{e1}$
    $) \wedge$
    $s1 \neq s2 \wedge$
    $m1 \geq m2$
    $)$
    $\Rightarrow$
    $(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq guard < 8 \wedge$
    $s1 = sign.(g1 + g2) \wedge$
    $0 \leq m1 - m2 < 2^{40} \wedge$
    $guard = 1 \wedge$
    $m1 - m2 > 2^{38} + 2^{37} \wedge (m1 - m2 - 1/4) \times 2^e < |g1 + g2| < (m1 - m2) \times 2^e$
    $)$

12. one out of 8 cases

    $(P \wedge e = e1 = e2 \wedge (P_1 \vee P_1{}') \wedge guard = 1 \wedge s1 \neq s2 \wedge m1 < m2)$
    $\Rightarrow$
    $(P \wedge P_4)\,[m/m2 - m1, s/s2]$

    analogous

13. one out of 7 cases

$(P \wedge P_4 \wedge guard = 1)$
$\Rightarrow$
$(P \wedge P_5)\,[m/m-1, guard/8-guard]$

i.e.

$(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq guard < 8 \wedge$
$s = sign.(g1+g2) \wedge$
$0 \leq m < 2^{40} \wedge$
$guard = 1 \wedge m > 2^{38} + 2^{37} \wedge (m - 1/4) \times 2^e < |g1+g2| < m \times 2^e$
$)$
$\Rightarrow$
$(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq 8 - guard < 8 \wedge$
$s = sign.(g1+g2) \wedge$
$0 \leq m - 1 < 2^{40} \wedge$
$8 - guard = 7 \wedge m - 1 \geq 2^{38} + 2^{37} \wedge$
$(m - 1 + 3/4) \times 2^e < |g1+g2| < (m - 1 + 1) \times 2^e$
$)$

14. $(P \wedge P_4 \wedge guard = 0)$
$\Rightarrow$
$(P \wedge P_5)$

i.e.

$(P \wedge$
$s = sign.(g1+g2) \wedge$
$0 \leq m < 2^{40} \wedge$
$guard = 0 \wedge m \times 2^e = |g1+g2|$
$)$
$\Rightarrow$
$(P \wedge$
$s = sign.(g1+g2) \wedge$
$0 \leq m < 2^{40} \wedge$
$guard = 0 \wedge m \times 2^e = |g1+g2|$
$)$

15. two out of 8 cases

15a. $(P \wedge P_5 \wedge m < 2^{39} \wedge guard = 4$
$\Rightarrow$
$(P \wedge P_6)\,[m/2 \times m + guard \div 4, e/e - 1, guard/(guard \bmod 4) \times 2]$
i.e.
$(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq guard < 8 \wedge$
$s = sign.(g1 + g2) \wedge$
$0 \leq m < 2^{40} \wedge$
$guard = 4 \wedge m \geq 1 \wedge (m + 1/2) \times 2^e = |g1 + g2| \wedge$
$m < 2^{39}$
$)$
$\Rightarrow$
$(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq (guard \bmod 4) \times 2 < 8 \wedge$
$s = sign.(g1 + g2) \wedge$
$0 \leq 2 \times m + guard \div 4 < 2^{40} \wedge$
$((guard \bmod 4) \times 2 = 0 \vee 2 \times m + guard \div 4 \geq 2^{39}) \wedge$
$even.((guard \bmod 4) \times 2) \wedge$
$(guard \bmod 4) \times 2 = 0 \wedge (2 \times m + guard \div 4) \times 2^{e-1} = |g1 + g2|$
$)$

15b. $(P \wedge P_5 \wedge m < 2^{39} \wedge guard = 7)$
$\Rightarrow$
$(P \wedge P_6)\,[m/2 \times m + guard \div 4, e/e - 1, guard/(guard \bmod 4) \times 2]$
i.e.
$(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq guard < 8 \wedge$
$s = sign.(g1 + g2) \wedge$
$0 \leq m < 2^{40} \wedge$
$guard = 7 \wedge m \geq 2^{38} + 2^{37} \wedge (m + 3/4) \times 2^e < |g1 + g2| < (m + 1) \times 2^e \wedge$
$m < 2^{39}$
$)$
$\Rightarrow$
$(0 \leq m1 < 2^{40} \wedge 0 \leq m2 < 2^{40} \wedge 0 \leq (guard \bmod 4) \times 2 < 8 \wedge$
$s = sign.(g1 + g2) \wedge$
$0 \leq 2 \times m + guard \div 4 < 2^{40} \wedge$
$((guard \bmod 4) \times 2 = 0 \vee 2 \times m + guard \div 4 \geq 2^{39})) \wedge$
$even.((guard \bmod 4) \times 2) \wedge$
$(guard \bmod 4) \times 2 = 6 \wedge$
$\quad (2 \times m + guard \div 4 + 1/2) \times 2^{e-1} < |g1 + g2| < (2 \times m + guard \div 4 + 1) \times 2^{e-1}$
$)$

16. $(P \wedge P_5 \wedge m \geq 2^{39})$
$\Rightarrow$
$(P \wedge P_3)$

i.e.

$(P \wedge$
$s = sign.(g1 + g2) \wedge$
$0 \leq m < 2^{40} \wedge$
case *guard* **of**
$0 : \qquad\qquad\qquad m \times 2^e = |g1 + g2|$
$1 : m \geq 2^{38} + 2^{37} \wedge m \times 2^e < |g1 + g2| < (m + 1/4) \times 2^e$
$2 : m \geq 2^{38} \qquad \wedge (m + 1/4) \times 2^e = |g1 + g2|$
$3 : m \geq 2^{38} + 2^{37} \wedge (m + 1/4) \times 2^e < |g1 + g2| < (m + 1/2) \times 2^e$
$4 : m \geq 1 \qquad\quad \wedge (m + 1/2) \times 2^e = |g1 + g2|$
$5 : m \geq 2^{38} + 2^{37} \wedge (m + 1/2) \times 2^e < |g1 + g2| < (m + 3/4) \times 2^e$
$6 : m \geq 2^{38} \qquad \wedge (m + 3/4) \times 2^e = |g1 + g2|$
$7 : m \geq 2^{38} + 2^{37} \wedge (m + 3/4) \times 2^e < |g1 + g2| < (m + 1) \times 2^e$
end case $\wedge$
$m \geq 2^{39}$
)
$\Rightarrow$
$(P \wedge$
$s = sign.(g1 + g2) \wedge$
$0 \leq m < 2^{41} \wedge$
$(guard = 0 \vee m \geq 2^{39}) \wedge$
case *guard* **of**
$0 : m \times 2^e = |g1 + g2|$
$1 : m \times 2^e < |g1 + g2| < (m + 1/4) \times 2^e$
$2 : (m + 1/4) \times 2^e = |g1 + g2|$
$3 : (m + 1/4) \times 2^e < |g1 + g2| < (m + 1/2) \times 2^e$
$4 : (m + 1/2) \times 2^e = |g1 + g2|$
$5 : (m + 1/2) \times 2^e < |g1 + g2| < (m + 3/4) \times 2^e$
$6 : (m + 3/4) \times 2^e = |g1 + g2|$
$7 : (m + 3/4) \times 2^e < |g1 + g2| < (m + 1) \times 2^e$
end case $\wedge$
$m < 2^{40}$
)

17. $(P \wedge (P_3 \vee P_6) \wedge \mathit{guard} \geq 4 \wedge m = 2^{40} - 1)$
$\Rightarrow$
$(P \wedge P_7)\,[m/2^{39}, e/e+1)]$

i.e.

$(P \wedge$
$s = \mathit{sign}.(g1 + g2) \wedge$
$0 \leq m < 2^{41} \wedge$
$(\mathit{guard} = 0 \vee m \geq 2^{39}) \wedge$
$(\textbf{case } \mathit{guard} \textbf{ of}$
$0 : m \times 2^e = |g1 + g2|$
$1 : m \times 2^e < |g1 + g2| < (m + 1/4) \times 2^e$
$2 : (m + 1/4) \times 2^e = |g1 + g2|$
$3 : (m + 1/4) \times 2^e < |g1 + g2| < (m + 1/2) \times 2^e$
$4 : (m + 1/2) \times 2^e = |g1 + g2|$
$5 : (m + 1/2) \times 2^e < |g1 + g2| < (m + 3/4) \times 2^e$
$6 : (m + 3/4) \times 2^e = |g1 + g2|$
$7 : (m + 3/4) \times 2^e < |g1 + g2| < (m + 1) \times 2^e$
$\textbf{end case} \wedge$
$m < 2^{40}$
$\vee$
$s = \mathit{sign}.(g1 + g2) \wedge$
$0 \leq m < 2^{40} \wedge$
$(\mathit{guard} = 0 \vee m \geq 2^{39}) \wedge$
$\mathit{even.guard} \wedge$
$\textbf{case } \mathit{guard} \textbf{ of}$
$0 : m \times 2^e = |g1 + g2|$
$2 : m \times 2^e < |g1 + g2| < (m + 1/2) \times 2^e$
$4 : (m + 1/2) \times 2^e = |g1 + g2|$
$6 : (m + 1/2) \times 2^e < |g1 + g2| < (m + 1) \times 2^e$
$\textbf{end case}$
$) \wedge$
$\mathit{guard} \geq 4 \wedge$
$m = 2^{40} - 1$
$)$
$\Rightarrow$
$(P \wedge$
$s = \mathit{sign}.(g1 + g2) \wedge$
$0 \leq 2^{39} < 2^{40} \wedge$
$(2^{39} \geq 2^{39} \vee 2^{39} \times 2^{e+1} = |g1 + g2|) \wedge$
$(2^{39} - 1/2) \times 2^{e+1} \leq |g1 + g2| < (2^{39} + 1/2) \times 2^{e+1} \wedge$
$(2^{39} = 2^{39} \rightarrow (2^{39} - 1/4) \times 2^{e+1} \leq |g1 + g2|)$
$)$

$\{\mathit{guard} \geq 4$ implies $(m + 1/2) \times 2^e \leq |g1 + g2| < (m + 1) \times 2^e$,
or, with $m = 2^{40} - 1$, $(2^{40} - 1/2) \times 2^e \leq |g1 + g2| < 2^{40} \times 2^e\}$

18. $(P \wedge (P_3 \vee P_6) \wedge \mathit{guard} \geq 4 \wedge m < 2^{40} - 1)$
$\Rightarrow$
$(P \wedge P_7)\ [m/m+1]$

i.e.

$(P \wedge$
$s = \mathit{sign}.(g1 + g2) \wedge$
$0 \leq m < 2^{41} \wedge$
$(\mathit{guard} = 0 \vee m \geq 2^{39}) \wedge$
$(\textbf{case}\ \mathit{guard}\ \textbf{of}$
$0 : m \times 2^e = |g1 + g2|$
$1 : m \times 2^e < |g1 + g2| < (m + 1/4) \times 2^e$
$2 : (m + 1/4) \times 2^e = |g1 + g2|$
$3 : (m + 1/4) \times 2^e < |g1 + g2| < (m + 1/2) \times 2^e$
$4 : (m + 1/2) \times 2^e = |g1 + g2|$
$5 : (m + 1/2) \times 2^e < |g1 + g2| < (m + 3/4) \times 2^e$
$6 : (m + 3/4) \times 2^e = |g1 + g2|$
$7 : (m + 3/4) \times 2^e < |g1 + g2| < (m + 1) \times 2^e$
$\textbf{end case} \wedge$
$m < 2^{40}$
$\vee$
$s = \mathit{sign}.(g1 + g2) \wedge$
$0 \leq m < 2^{40} \wedge$
$(\mathit{guard} = 0 \vee m \geq 2^{39}) \wedge$
$\mathit{even}.\mathit{guard} \wedge$
$\textbf{case}\ \mathit{guard}\ \textbf{of}$
$0 : m \times 2^e = |g1 + g2|$
$2 : m \times 2^e < |g1 + g2| < (m + 1/2) \times 2^e$
$4 : (m + 1/2) \times 2^e = |g1 + g2|$
$6 : (m + 1/2) \times 2^e < |g1 + g2| < (m + 1) \times 2^e$
$\textbf{end case}$
$) \wedge$
$\mathit{guard} \geq 4 \wedge$
$m < 2^{40} - 1$
$)$
$\Rightarrow$
$(P \wedge$
$s = \mathit{sign}.(g1 + g2) \wedge$
$0 \leq m + 1 < 2^{40} \wedge$
$(m + 1 \geq 2^{39} \vee (m + 1) \times 2^e = |g1 + g2|) \wedge$
$(m + 1 - 1/2) \times 2^e \leq |g1 + g2| < (m + 1 + 1/2) \times 2^e \wedge$
$(m + 1 = 2^{39} \rightarrow (m + 1 - 1/4) \times 2e \leq |g1 + g2|)$
$)$

$\{\mathit{guard} \geq 4$, hence $m \geq 2^{39} \wedge (m + 1/2) \times 2^e \leq |g1 + g2| < (m + 1) \times 2^e\}$

19. $(P \wedge (P_3 \vee P_6) \wedge guard < 4)$
$\Rightarrow$
$(P \wedge P_7)$

i.e.

$(P \wedge$
$s = sign.(g1 + g2) \wedge$
$0 \leq m < 2^{41} \wedge$
$(guard = 0 \vee m \geq 2^{39}) \wedge$
(**case** *guard* **of**
$0 : m \times 2^e = |g1 + g2|$
$1 : m \times 2^e < |g1 + g2| < (m + 1/4) \times 2^e$
$2 : (m + 1/4) \times 2^e = |g1 + g2|$
$3 : (m + 1/4) \times 2^e < |g1 + g2| < (m + 1/2) \times 2^e$
$4 : (m + 1/2) \times 2^e = |g1 + g2|$
$5 : (m + 1/2) \times 2^e < |g1 + g2| < (m + 3/4) \times 2^e$
$6 : (m + 3/4) \times 2^e = |g1 + g2|$
$7 : (m + 3/4) \times 2^e < |g1 + g2| < (m + 1) \times 2^e$
end case $\wedge$
$m < 2^{40}$
$\vee$
$s = sign.(g1 + g2) \wedge$
$0 \leq m < 2^{40} \wedge$
$(guard = 0 \vee m \geq 2^{39}) \wedge$
$even.guard \wedge$
case *guard* **of**
$0 : m \times 2^e = |g1 + g2|$
$2 : m \times 2^e < |g1 + g2| < (m + 1/2) \times 2^e$
$4 : (m + 1/2) \times 2^e = |g1 + g2|$
$6 : (m + 1/2) \times 2^e < |g1 + g2| < (m + 1) \times 2^e$
end case
$) \wedge$
$guard < 4)$
$)$
$\Rightarrow$
$(P \wedge$
$s = sign.(g1 + g2) \wedge$
$0 \leq m < 2^{40} \wedge$
$(m \geq 2^{39} \vee m \times 2^e = |g1 + g2|) \wedge$
$(m - 1/2) \times 2^e \leq |g1 + g2| < (m + 1/2) \times 2^e \wedge$
$(m = 2^{39} \rightarrow (m - 1/4) \times 2e \leq |g1 + g2|)$
$)$
$\{guard < 4$ implies $m \times 2^e \leq |g1 + g2| < (m + 1/2) \times 2^e\}$

Appendix B: the square–root test

In this appendix we present an ALGOL 60 program, originally written to test the square root implementation in the ALGOL 60 system for the EL X8. It showed that implementation correct, but also demonstrated the instability of the hardware implementation of the floating–point operations of the EL X8.

Here it is used to test the emulation of the floating–point hardware as described in this report.

First some theory.

For non–zero floating–point number x its square root is approximated in the following way. Write x as $a \times 2^{2\times b}$, with $1/4 \leq a < 1$. The $\sqrt{x} = \sqrt{a} \times 2^b$.

For $\sqrt{a}$ use approximation x_3 defined by:

$x_0 = .365681 + (5/8) \times a$

$x_1 = (a/x_0 + x_0)/2$

$x_2 = (a/x_1 + x_1)/2$

$x_3 = (a/x_2 + x_2)/2$

(Defining the relative error ϵ_i in x_i by setting $x_i = \sqrt{a}\times(1+\epsilon_i)$, we have $\epsilon_{i+1} = {\epsilon_i}^2/(1+\epsilon_i)$.)

In order to compute the truncation error (halting at x_3) we have to find the maximal relative error in x_0 in the interval $[1/4, 1)$ The two boundary errors are: for $a = 1/4$: $(.365681 + 5/32 - 1/2) \times 2 = .043862$, and for $a = 1$: $.365681 + 5/8 - 1 = -.009319$. The maximal relative interior error occurs where $\frac{d}{da}((.365681 + (5/8) \times a)/\sqrt{a} - 1) = 0$, i.e. for $a = (8/5) \times .365681 = .58509$, where $(.365681 + (5/8) \times a)/\sqrt{(a)} - 1 = -.043861$. This leads to a maximal relative error in x_3 of 1.28×10^{-13}. Since in the given interval $1/2 \leq \sqrt{a} < 1$ or $2^{39} \leq 2^{40} \times \sqrt{a} < 2^{40}$, the maximal absolute error is $2^{40} \times 1.28 \times 10^{-13}$, i.e. 0.141 bit.

To this we have to add the maximal rounding error. Since rounding errors in x_0, x_1, and x_2 are effectively removed in the next iteration step, only the rounding errors in the computation of x_3 do matter. The computation of a/x_2 gives an error of at most 0.5 bit. In the sum of a/x_2 and $x2$ (two numbers that are equal in the first 20 bits or so) the error in a/x_2 contributes only half, but we get an addition error of at most 0.5 bit. The division by two leads to no further errors: the only effect is to decrement the binary exponent by one. So we end with a maximal rounding error of 0.75 bit. Added to the maximal truncation error this gives an maximal absolute error of 0.90 bit.

Consequently, if the square root of a number can be represented exactly by an X8 floating–point number, the result is exact, since another result would differ at least 1 bit from the exact result, which is more than the maximal error. Therefore, the square roots of all the numbers of the form n^2 with $1 \leq n < 2^{20}$ are exact. We used that fact in a test program, originally meant to test the square root implementation, but later frequently used as a test program for the floating–point hardware! The run time of the test was some 13 minutes.

Here follows the program and its execution results.

```
   1   'begin' 'comment' square root test;
   2
   3         'integer' i; 'real' si;
   4
   5         'for' i:= 1 'step' 1 'until' 1 048 576 + 3 'do'
   6         'begin' si:= sqrt(i*i);
   7                 'if' si |= i
   8                 'then' 'begin' ABSFIXT(7,0,i); FLOT(13,3,si); NLCR 'end';
   9                 'if' i _: 100 000 * 100 000 = i
  10                 'then' 'begin' ABSFIXT(7,0,i); PRINTTEXT('('passed')');
  11                                NLCR
  12                        'end'
  13         'end'
  14   'end'
  15
```

```
  100000 passed
  200000 passed
  300000 passed
  400000 passed
  500000 passed
  600000 passed
  700000 passed
  800000 passed
  900000 passed
 1000000 passed
 1048577 +.1048577000002%+  7
 1048579 +.1048579000002%+  7
```

```
execution data
number of instructions executed:      110 965 581
execution time (seconds):                     772
average instruction time (microsec):            6.96
```

```
profile
linenumber  count       time     %
     5  18874540   99615538.75  12.9
     6  52428950  410037156.25  53.1
     7   6291474   27525208.75   3.6
     8      3800      22787.50   0.0
     9  33354412  234689332.50  30.4
    10      9339      49433.75   0.0
    11       380       1637.50   0.0
```

References

[1] E.W. Dijkstra. *Guarded commands, nondeterminacy and formal derivation of programs.*
Comm. ACM 18 (1975) 453.

[2] E.W. Dijkstra. *A Discipline of Programming.*
Prentice–Hall,1976.

[3] A.A. Grau. *On a floating–point representation for use with algorithmic languages.*
Comm. ACM 5 (1962) 160.

[4] C.A.R. Hoare. *An Axiomatic Basis for Computer Programming.*
Comm. ACM 12 (1969) 576.