Innocuous Double Rounding of Basic Arithmetic Operations

Pierre Roux
ISAE, ONERA

Double rounding occurs when a floating-point value is first rounded to an intermediate precision before being rounded to a final precision. The result of two such consecutive roundings can differ from the result obtained when directly rounding to the final precision. Double rounding practically happens, for instance, when implementing the IEEE754 binary32 format with an arithmetic unit performing operations only in the larger binary64 format, such as done in the PowerPC or x87 floating-point units. It belongs to the folklore in the floating-point arithmetic community that double rounding is innocuous for the basic arithmetic operations (addition, division, multiplication, and square root) as soon as the final precision is about twice larger than the intermediate one. This paper adresses the formal proof of this fact considering underflow cases and its extension to radices other than two.

1. INTRODUCTION

Floating-point numbers are commonly used to efficiently perform numerical computations, which are then performed with a bounded *precision*. That is, only a finite number of bits are kept after each arithmetic operation. Multiple choices are available for this precision, typical examples being 24 bits for the IEEE754 binary32 format and 53 for the binary64 format.

Double rounding occurs when a value is first rounded to an intermediate precision before being rounded to a final precision. The result of two such consecutive roundings can differ from the result obtained when directly rounding to the final precision. In radix 10, for instance, 1.495 could be rounded to 1.50 then to 2 if first rounding to a 3-digit precision then to one digit whereas a direct rounding to the nearest value with one digit would give 1. Double rounding practically happens, for instance, when implementing the IEEE754 binary32 [IEE08] format with an arithmetic unit only performing operations in the larger binary64 format, such as done in the PowerPC or x87 floating-point units.

It belongs to the folklore in the floating-point arithmetic community that double rounding is innocuous for the basic arithmetic operations (addition, subtraction, multiplication, division and square root) as soon as the final precision is about twice larger than the intermediate one. The IEEE754 binary32 and binary64 formats [IEE08] fulfill this condition. Figueroa [Fig95] published pen-and-paper proofs of these results but only considering radix 2 and completely ignoring underflows. We try to give more general results in this paper. Moreover, such proofs tend to involve quite a number of rather subtle corner cases which make them particularly error prone. We therefore use Coq [BCHPM04, Coq12] to ensure no such corner case has been missed. The impact of double rounding on some particular floating-point al-

This work was done while the author was a visiting researcher at LRI, Inria Saclay – Île-de-France.

gorithms such as the Veltkamp/Dekker's algorithms has already been studied using Coq [MDMM13], but its innocuousness, under some conditions, on basic arithmetic operations is only briefly mentioned.

More formally, given a floating-point format $\mathbb{F}$, a rounding $\circ_1$ in this format and a second rounding $\circ_2$ in a larger-precision format, we want to ensure:

$$\forall x, y \in \mathbb{F}, \circ_1(\circ_2(x \diamond y)) = \circ_1(x \diamond y)$$

under some conditions, for $\diamond \in \{+, -, \times, /, \sqrt{}\}$. That is, the second rounding $\circ_2$ is innocuous for basic arithmetic operations. This paper aims at studying under which conditions the previous holds and to formally prove such results using a proof assistant (Coq). It is worth noting that the results are highly dependent on the operation $\diamond$ and that they may not hold for operations other than $\{+, -, \times, /, \sqrt{}\}$. Since all the proofs presented in the remaining of the paper have been formally verified within Coq, we only present proof sketches. Although not all subcases are precisely demonstrated, enough is shown to give a good idea of what is going on.

The case of directed roundings (toward $-\infty$ or $+\infty$ or roundings to zero or away) is rather simple: the above result holds if and only if $\circ_1$ and $\circ_2$ are rounding in the same direction. The remaining will therefore focus on the case where $\circ_1$ and $\circ_2$ are roundings to nearest (possibly with different tie-break rules). We can nevertheless notice that the results still hold (under possibly weaker hypotheses) if $\circ_1$ is a directed rounding and $\circ_2$ a rounding to nearest. The converse case ($\circ_1$ to nearest and $\circ_2$ directed) however usually requires slightly stronger hypothesis.

Finally, we do not deal with overflows since they are harmless as soon as any number which can be represented in the smaller precision format $\mathbb{F}$ can also be represented in the larger precision, which is a reasonable assumption on exponent ranges.

The remainder of this section introduces the floating-point formalisms used in our proofs. Then, Sections 2, 3, 4, and 5 illustrate our proofs about respectively the multiplication, the addition/subtraction, the square root and the division. For the sake of clarity, the proofs are presented on a formalism not modeling underflows, although our actual Coq proofs handle them and precise requirements on underflow will be given. Eventually, Section 6 summarizes the results and concludes.

All our Coq developments are part of the Flocq library [BM11] available[1] at http://flocq.gforge.inria.fr/.

1.1 Floating-point Formats

We first formally define some floating-point formats used throughout the remaining of the paper. A generic format is first defined, then particular cases: a fixed precision format without underflow, a format with gradual underflows and one with abrupt underflows. All these formats are formally defined in the Coq library for floating-point arithmetic Flocq [BM11].

For this purpose, we need a discrete logarithm.

[1] In files `src/appli/Fappli_double_round.v` and `examples/Double_round_beta_odd.v`, starting with version 2.4.0 of Flocq.

Definition 1. Given $\beta \in \mathbb{Z}, \beta \geq 2$, we define the discrete logarithm $\ln_\beta(x)$ for radix β of a real number $x \in \mathbb{R} \setminus \{0\}$ as the unique value e_x in $\mathbb{Z}$ satisfying

$$\beta^{e_x - 1} \leq |x| < \beta^{e_x}.$$

A generic format $\mathbb{F}_\varphi$ is defined by a function $\varphi : \mathbb{Z} \to \mathbb{Z}$ giving, for all numbers sharing the same discrete logarithm, the precision they must be encoded with.

Definition 2. Given $\varphi : \mathbb{Z} \to \mathbb{Z}$, a value $x \in \mathbb{R}$ is said to be in the generic format $\mathbb{F}_\varphi$ if $x = 0$ or if there is some $m \in \mathbb{Z}$ such that

$$x = m\, \beta^{\varphi(\ln_\beta(x))}.$$

m is then called the *mantissa* of x and $\varphi(\ln_\beta(x))$ its *canonical exponent*. In the Flocq library, the proposition `generic_format` $\beta\ \varphi\ x$ means that $x \in \mathbb{F}_\varphi$.

This generic format then allows us to define more concrete formats. Given some $p \in \mathbb{Z}$, $p \geq 1$, the format FLX_p models precision p floating-point numbers without underflow.

Definition 3. Given $p \in \mathbb{Z}$, $p \geq 1$, the format FLX_p is defined as $\mathbb{F}_\varphi$ with $\varphi : e \mapsto e - p$ (denoted `FLX_exp` p in Flocq).

Although this format remains mostly theoretical since hardware floating-point numbers can underflow, it is still a good model of what happens with actual floating-point values when no underflow occurs.

Given $p \in \mathbb{Z}$, $p \geq 1$, and $e_{min} \in \mathbb{Z}$, the format $\mathrm{FLT}_{p,e_{min}}$ models precision p floating-point numbers with underflow handled by subnormals of exponent e_{min}.

Definition 4. Given $p, e_{min} \in \mathbb{Z}$, $p \geq 1$, the format $\mathrm{FLT}_{p,e_{min}}$ is defined as $\mathbb{F}_\varphi$ with $\varphi : e \mapsto \max(e - p, e_{min})$ (denoted `FLT_exp` $e_{min}\ p$ in Flocq).

Remark 5. $\mathrm{FLT}_{24,-149}$ *with radix* $\beta = 2$ *corresponds to the IEEE754 binary32 format and* $\mathrm{FLT}_{53,-1074}$ *to binary64 (neglecting overflows).*

Finally, given $p \in \mathbb{Z}$ and $e_{min} \in \mathbb{Z}$, the format $\mathrm{FTZ}_{p,e_{min}}$ models precision p floating-point numbers with underflow handled by flushing either to zero or to the smallest normal number.

Definition 6. Given $p, e_{min} \in \mathbb{Z}$, the format $\mathrm{FTZ}_{p,e_{min}}$ is defined as $\mathbb{F}_\varphi$ with $\varphi : e \mapsto e - p$ when $e \geq e_{min} + p$ and $e \mapsto e_{min} + p - 1$ otherwise (denoted `FTZ_exp` $e_{min}\ p$ in Flocq).

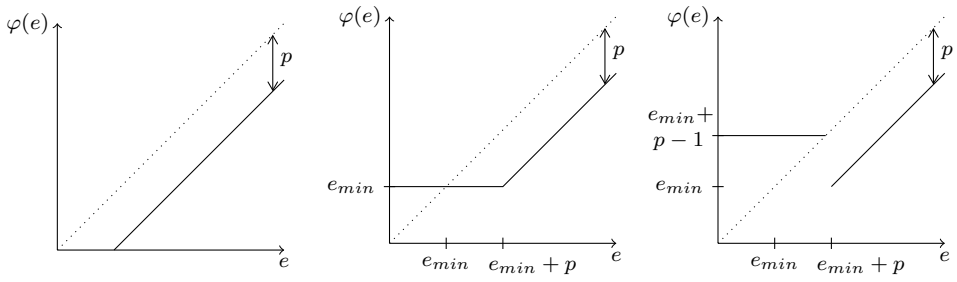
The φ functions defining FLX, FLT and FTZ formats are illustrated on Figure 1.

It only remains to define roundings.

Definition 7. A *rounding* on a floating point format $\mathbb{F}$ is a function $\circ : \mathbb{R} \to \mathbb{F}$ monotone and equal to the identity on $\mathbb{F}$, that is

$$\forall x, y \in \mathbb{R}, x \leq y \Rightarrow \circ(x) \leq \circ(y)$$
$$\forall f \in \mathbb{F}, \circ(f) = f.$$

This corresponds to the hypothesis `Valid_rnd rnd` in Flocq while the rounding function $\circ$ on $\mathbb{F}_\varphi$ is denoted by `round` $\beta\ \varphi$ `rnd`.



(a) FLX_p (b) $\mathrm{FLT}_{p,e_{min}}$ (c) $\mathrm{FTZ}_{p,e_{min}}$

Fig. 1: Illustration of the functions φ defining the formats FLX, FLT and FTZ. The difference $(e - \varphi(e))$ between the diagonal (doted line) and the graph of φ is the size of the mantissa for numbers of magnitude e in format $\mathbb{F}_\varphi$ (if non positive, numbers can only be encoded as 0). These functions are the same for normal numbers ($e \geq e_{min} + p$) and only differ for subnormal numbers. FLX doesn't model them while FLT has gradual underflow and FTZ abrupt underflow.

Definition 8. A *rounding to nearest* is a rounding $\circ$ that returns a nearest value in $\mathbb{F}$:

$$\forall x \in \mathbb{R}, \forall f \in \mathbb{F}, |x - \circ(x)| \leq |x - f|\,.$$

They are of the form `round` β φ (`Znearest` _) in Flocq.

Remark 9. *In case x lies halfway between two consecutive values in $\mathbb{F}$, $\circ(x)$ is not uniquely defined as it can be any of these two consecutive values. This choice is called a* tie-break rule. *Since most of our proofs are valid regardless of the tie-break rule (i.e., for* any *rounding to nearest) we don't extend on this point.*

2. MULTIPLICATION

Proofs for the addition/subtraction being rather involved, let us begin with the multiplication for which proofs are much simpler.

2.1 General Case

Figueroa [Fig95] proved double rounding to be innocuous for the multiplication if it is performed with a precision at least twice as large. Although his proof only addressed radix 2, this holds for any radix.

Theorem 10. *([Fig95], `double_round_mult_FLX` in our Coq development) For $p_1, p_2 \in \mathbb{Z}$, if $p_2 \geq 2p_1$, then for $\circ_1$ and $\circ_2$ any roundings (for instance, toward $-\infty$ or to nearest with any tie) respectively in FLX_{p_1} and FLX_{p_2}:*

$$\forall x, y \in \mathrm{FLX}_{p_1}, \circ_1(\circ_2(x \times y)) = \circ_1(x \times y)\,.$$

This in particular applies to binary32/64 when no underflow occurs.

Remark 11. *[Fig95] This bound $2p_1$ is optimal as shown by the following counterexample. With radix $\beta = 2$, for $p_1 = 4, p_2 = 7$, we have for $x = y = 1.101_2$*

$$\circ_1(\circ_2(x \times y)) = \circ_1(\circ_2(10.101001_2)) = \circ_1(10.10100_2) = 10.10_2$$

which differs from $\circ_1(10.101001_2) = 10.11_2$ for $\circ_1$ and $\circ_2$ appropriate roundings to nearest (for instance, the common rounding to nearest with tie-break to even).

PROOF. x and y having p_1 significand digits, $x \times y$ has at most $2p_1 \leq p_2$ digits hence $\circ_2(x \times y) = x \times y$. □

The previous result still applies to binary32/64 even in case of underflow. Indeed, proofs for the FLX, FLT and FTZ formats (`double_round_mult_{FLX,FLT,FTZ}` in our Coq development) are just corollaries of a single proof (`double_round_mult`) carried out on the generic format $\mathbb{F}_\varphi$. This general proof shows innocuity of double rounding of multiplication under the assumption

$$\forall e_x, e_y \in \mathbb{Z}, \varphi_2(e_x + e_y) \leq \varphi_1(e_x) + \varphi_1(e_y) \wedge \varphi_2(e_x + e_y - 1) \leq \varphi_1(e_x) + \varphi_1(e_y),$$

the main argument being the same than in the above proof.

2.2 Odd Radix

It is worth noting that a much better result can be obtained for odd radices β, since the hypothesis $p_2 \geq 2p_1$ in Theorem 10 can be replaced with the weaker $p_2 \geq p_1$ when β is odd. This nice result is, at least currently[2], certainly perfectly useless. We nevertheless chose to develop it for the sake of exhaustivity and considering it did not implied a huge overhead on our Coq development.

THEOREM 12. *(`double_round_mult_beta_odd_FLX`) When β is odd, for $p_1, p_2 \in \mathbb{Z}$, if $p_2 \geq p_1$, then for $\circ_1$ and $\circ_2$ roundings to nearest, with any tie, respectively in* FLX_{p_1} *and* FLX_{p_2}

$$\forall x, y \in \mathrm{FLX}_{p_1}, \circ_1(\circ_2(x \times y)) = \circ_1(x \times y)\,.$$

To give an idea of the proof, we first need a few additional definitions and lemmas.

Definition 13. $\circ_\varphi^\downarrow$ denotes *rounding towards* $-\infty$, i.e., $\circ_\varphi^\downarrow(x)$ is the largest floating-point value in $\mathbb{F}_\varphi$ below x:

$$\circ_\varphi^\downarrow : x \in \mathbb{R} \mapsto \max\{f \in \mathbb{F}_\varphi \mid f \leq x\}.$$

Definition 14. $\mathrm{ulp}_\varphi(x)$ denotes the *unit in last place* of x in format $\mathbb{F}_\varphi$:

$$\mathrm{ulp}_\varphi(x) := \beta^{\varphi(\ln_\beta(x))}.$$

Definition 15. $\mathrm{midp}_\varphi(x)$ denotes the following value, called *midpoint*

$$\mathrm{midp}_\varphi(x) := \circ_\varphi^\downarrow(x) + \frac{\mathrm{ulp}_\varphi(x)}{2}$$

as it lies halfway between two consecutive floating-point values around x.

LEMMA 16. *(`double_round_lt_mid_further_place`) Given two floating-point formats $\mathbb{F}_{\varphi_1}$ and $\mathbb{F}_{\varphi_2}$, two roundings to nearest, with any tie, $\circ_1$ and $\circ_2$ in these formats, for all $x \in \mathbb{R}$, if $x > 0$, $\varphi_2(\ln_\beta(x)) < \varphi_1(\ln_\beta(x))$, $\varphi_1(\ln_\beta(x)) \leq \ln_\beta(x)$ and*

$$x < \mathrm{midp}_{\varphi_1}(x) - \frac{\mathrm{ulp}_{\varphi_2}(x)}{2}$$

then

$$\circ_1(\circ_2(x)) = \circ_1(x)\,.$$

[2]There have been a couple of computers using a radix 3 arithmetic, e.g., Setun built in Moscow in the 60s and Ternac developped in New York in the 70s.

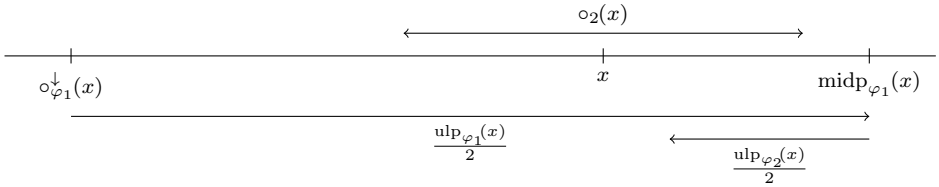


Fig. 2: Illustration of Lemma 16. Since x is below $\mathrm{midp}_{\varphi_1}(x)$, $\circ_1(x) = \circ_{\varphi_1}^{\downarrow}(x)$ and since x is below $\mathrm{midp}_{\varphi_1}(x) - \mathrm{ulp}_{\varphi_2}(x)/2$ then $\circ_2(x)$ is also below $\mathrm{midp}_{\varphi_1}(x)$, implying $\circ_1(\circ_2(x)) = \circ_{\varphi_1}^{\downarrow}(x)$.

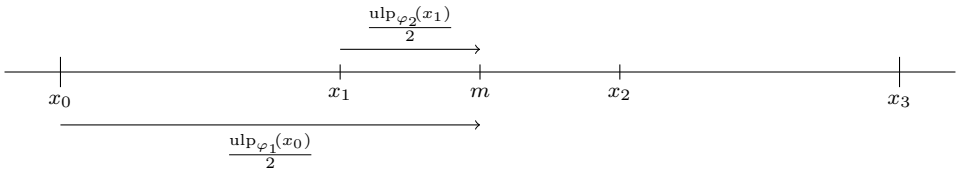


Fig. 3: Illustration of Lemma 17. In radix $\beta = 3$, with $\varphi_2 = \varphi_1 - 1$, x_0 and x_3 are consecutive values in $\mathbb{F}_{\varphi_1}$ while x_0, x_1, x_2 and x_3 are consecutive values in $\mathbb{F}_{\varphi_2}$. The midpoint m in $\mathbb{F}_{\varphi_1}$ is also a midpoint in $\mathbb{F}_{\varphi_2}$: $m = x_0 + \frac{\mathrm{ulp}_{\varphi_1}(x_0)}{2} = x_1 + \frac{\mathrm{ulp}_{\varphi_2}(x_1)}{2}$. This property relies on β being odd.

This is illustrated on Figure 2.

Proof. Since $\varphi_2(\ln_\beta(x)) \leq \varphi_1(\ln_\beta(x))$, we get $\circ_2(x) - \circ_{\varphi_1}^{\downarrow}(x) = \left|\circ_2(x) - \circ_{\varphi_1}^{\downarrow}(x)\right|$. Moreover, $\circ_2$ being a rounding to nearest, we get for all $y \in \mathbb{R}$, $|\circ_2(y) - y| \leq \frac{\mathrm{ulp}_{\varphi_2}(y)}{2}$ and the hypothesis $x < \mathrm{midp}_{\varphi_1}(x) - \frac{\mathrm{ulp}_{\varphi_2}(x)}{2}$ gives us $\left|x - \circ_{\varphi_1}^{\downarrow}(x)\right| < \frac{\mathrm{ulp}_{\varphi_1}(x) - \mathrm{ulp}_{\varphi_2}(x)}{2}$. Thus $\circ_2(x) - \circ_{\varphi_1}^{\downarrow}(x) \leq |\circ_2(x) - x| + \left|x - \circ_{\varphi_1}^{\downarrow}(x)\right| < \frac{\mathrm{ulp}_{\varphi_1}(x)}{2}$. $\circ_1$ being a rounding to nearest, $\circ_1(\circ_2(x))$ and $\circ_1(x)$ are then both equal to $\circ_{\varphi_1}^{\downarrow}(x)$. □

Contrary to the previous lemmas which were valid for any radix β, the following result shows a particularly interesting property of odd radices.

Lemma 17. *(`midpoint_beta_odd_remains`) When β is odd, for all $\varphi_1, \varphi_2 : \mathbb{Z} \to \mathbb{Z}$, for all $x \in \mathbb{F}_{\varphi_1}$, if $\varphi_2(\ln_\beta(x)) \leq \varphi_1(\ln_\beta(x))$, then there exists an $x' \in \mathbb{F}_{\varphi_2}$ such that $x + \frac{\mathrm{ulp}_{\varphi_1}(x)}{2} = x' + \frac{\mathrm{ulp}_{\varphi_2}(x)}{2}$.*

This result is illustrated on Figure 3. Intuitively, this means that, with odd radices, if a real number $x + \frac{\mathrm{ulp}_{\varphi_1}(x)}{2}$ is a midpoint for some precision, it is also a midpoint in any larger precision. This is proved by induction on $\varphi_1(\ln_\beta(x)) - \varphi_2(\ln_\beta(x)) \in \mathbb{N}$ and makes it possible to prove the following lemma.

Lemma 18. *(`neq_midpoint_beta_odd`) When β is odd, for all $\varphi_1, \varphi_2 : \mathbb{Z} \to \mathbb{Z}$, for all $x \in \mathbb{R}$, if $\varphi_2(\ln_\beta(x)) \leq \varphi_1(\ln_\beta(x))$ and $x \neq \mathrm{midp}_{\varphi_1}(x)$ then $\circ_1(\circ_2(x)) = \circ_1(x)$ for any rounding to nearest, with any tie, $\circ_1$ and $\circ_2$ respectively in $\mathbb{F}_{\varphi_1}$ and $\mathbb{F}_{\varphi_2}$.*

This means that with odd radices, double rounding is innocuous for all real values which are not midpoints.

PROOF. Lemma 16 concludes when $x < \mathrm{midp}_{\varphi_1}(x) - \frac{\mathrm{ulp}_{\varphi_2}(x)}{2}$. Let us then assume

$$\mathrm{midp}_{\varphi_1}(x) - \frac{\mathrm{ulp}_{\varphi_2}(x)}{2} \leq x < \mathrm{midp}_{\varphi_1}(x)\,.$$

From Lemma 17, there exist an $x' \in \mathbb{F}_{\varphi_2}$ such that $x' = \mathrm{midp}_{\varphi_1}(x) - \frac{\mathrm{ulp}_{\varphi_2}(x)}{2}$ and

$$x' \leq x < x' + \frac{\mathrm{ulp}_{\varphi_2}(x)}{2}$$

hence $\circ_2(x) = x'$. Since $\circ^{\downarrow}_{\varphi_1}(x) \leq x' < \mathrm{midp}_{\varphi_1}(x)$, then $\circ_1(x') = \circ^{\downarrow}_{\varphi_1}(x) = \circ_1(x)$. The proof is similar for $x > \mathrm{midp}_{\varphi_1}(x)$. □

A last lemma is then needed to prove the theorem.

LEMMA 19. *(`float_neq_midpoint_beta_odd`) When β is odd, for all $x \in \mathbb{R}$, $\varphi : \mathbb{Z} \to \mathbb{Z}$, if there exist $m, e \in \mathbb{Z}$ such that $x = m\beta^e$, then $x \neq \mathrm{midp}_{\varphi}(x)$*

PROOF. If $e \geq \varphi(\ln_\beta(x))$, then both x and $\circ^{\downarrow}_{\varphi}(x)$ are multiples of $\beta^{\varphi(\ln_\beta(x))}$ whereas $\frac{\mathrm{ulp}_{\varphi}(x)}{2}$ is not. Otherwise, Lemma 17 allows us to conclude by a similar reasoning. □

PROOF (THEOREM 12). Since $x \times y = (m_x \times m_y)\,\beta^{e_x+e_y}$, with m_x, m_y and e_x, e_y the mantissas and exponents of x and y, Lemmas 18 and 19 make it possible to conclude. □

3. ADDITION

3.1 General Case

Figueroa [Fig95] proved double rounding to be innocuous for the addition if it is performed with a precision at least strictly twice as large. Although his proof only addressed radix 2, this holds for any radix.

THEOREM 20. *[Fig95] (`double_round_plus_FLX`) For $p_1, p_2 \in \mathbb{Z}$, if $p_2 \geq 2p_1+1$, then for $\circ_1$ and $\circ_2$ roundings to nearest, with any tie, respectively in FLX_{p_1} and FLX_{p_2}:*

$$\forall x, y \in \mathrm{FLX}_{p_1}, \circ_1(\circ_2(x+y)) = \circ_1(x+y)\,.$$

This applies to IEEE754 binary32/64 when no underflow occurs.

REMARK 21. *[Fig95] This bound $2p_1+1$ is optimal as shown by the following counterexample. With radix $\beta = 2$, for $p_1 = 4, p_2 = 8$, we have for $x = 1.001_2$ and $y = 0.00001111_2$*

$$\circ_1(\circ_2(x+y)) = \circ_1(\circ_2(1.00101111_2)) = \circ_1(1.0011000_2) = 1.010_2$$

which differs from $\circ_1(1.00101111_2) = 1.001_2$ when $\circ_1$ and $\circ_2$ are roundings to nearest with tie-break to even.

We need the following lemma to prove the previous theorem.

LEMMA 22. *For all $\varphi : \mathbb{Z} \to \mathbb{Z}$, $x \in \mathbb{F}_\varphi$, $y \in \mathbb{R}$, if $x > 0$, $y > 0$ and $\ln_\beta(y) \leq \varphi(\ln_\beta(x)) - 2$, then*

$$0 < (x+y) - \circ^{\downarrow}_{\varphi}(x+y) < \beta^{\varphi(\ln_\beta(x))-2}.$$

PROOF. Since $\ln_\beta(y) \leq \varphi(\ln_\beta(x)) - 2$, we get $y < \mathrm{ulp}_\varphi(x) = \mathrm{ulp}_\varphi(x+y)$, hence $\circ^\downarrow_\varphi(x+y) = x$, that is $(x+y) - \circ^\downarrow_\varphi(x+y) = y$ which implies the result. □

PROOF (THEOREM 20). We will only consider the case $x > 0$, $y > 0$ and $y \leq x$, other cases being pretty similar or easy to deduce. Either $\ln_\beta(y) \geq \varphi_1(\ln_\beta(x)) - 1$ in which case $x+y$ has at most $2p_1+1$ significand digits, which means that $\circ_2(x+y) = x+y$, or $\ln_\beta(y) \leq \varphi_1(\ln_\beta(x)) - 2$ in which case the result follows from Lemmas 16 and 22. □

The previous result still applies to IEEE754 binary32/64 even in case of underflow (`double_round_plus_FLT`), with a similar proof thanks to the generic format $\mathbb{F}_\varphi$. Indeed, proofs for formats with or without underflow are derived as immediate corollaries of a proof on the generic format.

Under the assumption $\beta \geq 3$, the bound $2p_1 + 1$ can be replaced by $2p_1$ in Theorem 20. The proof is similar (`double_round_plus_beta_ge_3_FLX`).

REMARK 23. *This bound $2p_1$ is optimal as shown by the following counterexample. With radix $\beta = 10$, for $p_1 = 4, p_2 = 7$, we have for $x = 1.001_{10}$ and $y = 0.0004995_{10}$*

$$\circ_1(\circ_2(x+y)) = \circ_1(\circ_2(1.0014995_{10})) = \circ_1(1.001500_{10}) = 1.002_{10}$$

which differs from $\circ_1(1.0014995_{10}) = 1.001_{10}$ when $\circ_1$ and $\circ_2$ are roundings to nearest with tie-break to even.

Finally, we can notice that the same results can be immediately deduced for the subtraction.

3.2 Odd Radix

As for the multiplication, a way better bound is obtained for odd radices.

THEOREM 24. *(`double_round_plus_beta_odd_FLX`) When β is odd, for $p_1, p_2 \in \mathbb{Z}$, if $p_2 \geq p_1$, then for $\circ_1$ and $\circ_2$ roundings to nearest, with any tie, respectively in FLX_{p_1} and FLX_{p_2}:*

$$\forall x, y \in \mathrm{FLX}_{p_1}, \circ_1(\circ_2(x+y)) = \circ_1(x+y)\,.$$

PROOF. Denoting m_x, m_y and e_x, e_y the mantissas and exponents of x and y, $x+y = m\beta^e$ with $e = \min(e_x, e_y)$ and $m = m_x\beta^{e_x - \min(e_x,e_y)} + m_y\beta^{e_y - \min(e_x,e_y)} \in \mathbb{Z}$. Lemmas 18 and 19 make it possible to conclude. □

4. SQUARE ROOT

4.1 General Case

Figueroa [Fig95] proved double rounding to be innocuous for the square root if it is performed with a precision larger than twice the original precision plus two. Although his proof only addressed radix 2, this holds for any radix.

THEOREM 25. *[Fig95] (`double_round_sqrt_FLX`) For $p_1, p_2 \in \mathbb{Z}$, if $p_2 \geq 2p_1+2$, then for $\circ_1$ and $\circ_2$ roundings to nearest respectively in FLX_{p_1} and FLX_{p_2}*

$$\forall x \in \mathrm{FLX}_{p_1}, \circ_1\big(\circ_2\big(\sqrt{x}\big)\big) = \circ_1\big(\sqrt{x}\big)\,.$$

This applies to IEEE754 binary32/64 when no underflow occurs.

Remark 26. *[Fig95] This bound $2p_1+2$ is optimal as shown by the following counterexample. With radix $\beta=2$, for $p_1=4, p_2=9$, we have for $x=0.1111_2$*

$$\circ_1\big(\circ_2\big(\sqrt{x}\big)\big)=\circ_1(\circ_2(0.1111011111\ldots_2))=\circ_1(0.111110000_2)=1.000_2$$

which differs from $\circ_1(0.1111011111\ldots_2)=0.1111_2$ when $\circ_1$ and $\circ_2$ are roundings to nearest with tie-break to even.

Proof (Theorem 25). If $\sqrt{x}<\mathrm{midp}_{\varphi_1}(\sqrt{x})-\frac{\mathrm{ulp}_{\varphi_2}(\sqrt{x})}{2}$, the result follows from Lemma 16. Similarly, the result holds when $\sqrt{x}>\mathrm{midp}_{\varphi_1}(\sqrt{x})+\frac{\mathrm{ulp}_{\varphi_2}(\sqrt{x})}{2}$. We will now show that x cannot lie between those two bounds. Let us first denote $u_1:=\mathrm{ulp}_{\varphi_1}(\sqrt{x})$, $u_2:=\mathrm{ulp}_{\varphi_2}(\sqrt{x})$, $a:=\circ^{\downarrow}_{\varphi_1}(\sqrt{x})$, $b:=\frac{u_1-u_2}{2}$ and $b':=\frac{u_1+u_2}{2}$. Then assuming

$$a+b\le\sqrt{x}\le a+b'$$

we get

$$a^2+u_1a-u_2a+b^2\le x\le a^2+u_1a+u_2a+b'^2.$$

a^2+u_1a as well as x can be divided by u_1^2. However $-u_2a+b^2>0$ and $u_2a+b'^2<u_1^2$ which is absurd. □

The previous result still applies to IEEE754 binary32/64 even in case of underflow (`double_round_sqrt_FLT`).

Under the assumption $\beta\ge 4$, the bound $2p_1+2$ can be replaced by $2p_1+1$ in Theorem 25. The proof is similar (`double_round_sqrt_beta_ge_4_FLX`).

Remark 27. *This bound $2p_1+1$ is optimal as shown by the following counterexample. With radix $\beta=10$, for $p_1=4, p_2=8$, we have for $x=0.9999_{10}$*

$$\circ_1\big(\circ_2\big(\sqrt{x}\big)\big)=\circ_1(\circ_2(0.999949998\ldots_{10}))=\circ_1(0.99995000_{10})=1.000_{10}$$

which differs from $\circ_1(0.999949998\ldots_{10})=0.9999_{10}$ when $\circ_1$ and $\circ_2$ are roundings to nearest with tie-break to even.

4.2 Odd Radix

As for the multiplication and addition, a way better bound is obtained for odd radices.

Theorem 28. *(`double_round_sqrt_beta_odd_FLX`) When β is odd, for $p_1,p_2\in\mathbb{Z}$, if $p_2\ge p_1$, then for $\circ_1$ and $\circ_2$ roundings to nearest respectively in FLX_{p_1} and FLX_{p_2}:*

$$\forall x\in\mathrm{FLX}_{p_1},\circ_1\big(\circ_2\big(\sqrt{x}\big)\big)=\circ_1\big(\sqrt{x}\big)\,.$$

Proof. According to Lemma 18, it is enough to prove that $\sqrt{x}\ne\mathrm{midp}_{\varphi_1}(\sqrt{x})$. Assuming the contrary, we get

$$x=r^2+ru+\frac{u^2}{4}$$

where $r:=\circ^{\downarrow}_{\varphi_1}(\sqrt{x})$ and $u:=\mathrm{ulp}_{\varphi_1}(\sqrt{x})$. Then, when $\varphi_1(\ln_\beta(x))\ge 2\varphi_1(\ln_\beta(\sqrt{x}))$, we have x, r^2 and ru multiples of u^2 whereas $\frac{u^2}{4}$ is not. Otherwise, Lemma 17 makes it possible to conclude with a similar reasoning. □

5. DIVISION

5.1 Even Radix

Figueroa [Fig95] proved double rounding to be innocuous for the division if it is performed with a precision at least twice as large. Although his proof only addressed radix 2, this holds for any *even* radix.

THEOREM 29. *[Fig95] (**double_round_div_FLX**) When β is even, for $p_1, p_2 \in \mathbb{Z}$, if $p_2 \geq 2p_1$, then for $\circ_1$ and $\circ_2$ roundings to nearest, with any tie, respectively in* FLX_{p_1} *and* FLX_{p_2}*:*

$$\forall x, y \in \mathrm{FLX}_{p_1}, y \neq 0 \Rightarrow \circ_1(\circ_2(x/y)) = \circ_1(x/y)\,.$$

This applies to IEEE754 binary32/64 when no underflow occurs.

REMARK 30. *[Fig95] This bound $2p_1$ is optimal as shown by the following counterexample. With radix $\beta = 2$, for $p_1 = 4, p_2 = 7$, we have for $x = 1.000_2$ and $y = 0.1111_2$*

$$\circ_1(\circ_2(x/y)) = \circ_1(\circ_2(1.000100010\ldots_2)) = \circ_1(1.0001000_2) = 1.000_2$$

which differs from $\circ_1(1.000100010\ldots_2) = 1.001_2$ when $\circ_1$ and $\circ_2$ are roundings to nearest with tie-break to even.

PROOF. Denoting $\mathrm{midp}_{\varphi_1}(x/y)$ with m and $\frac{\mathrm{ulp}_{\varphi_2}(x/y)}{2}$ with u, from Lemma 16, the result holds when $x/y < m - u$ and similarly when $x/y > m + u$. When $m - u \leq x/y < m$, an argument similar to the one used in proof of Theorem 25 allows to conclude and similarly when $m < x/y \leq m + u$. Finally, when $x/y = m$, since β is even, $x/y \in \mathbb{F}_{\varphi_2}$ as soon as $\varphi_2(x/y) < \varphi_1(x/y)$ which is implied by $p_2 \geq 2p_1$ and $p_1 \geq 1$. □

The previous result still applies to IEEE754 binary32/64 even in case of underflow (`double_round_div_FLT`).

5.2 Odd Radix

The result does not hold for odd radices.

REMARK 31. *The result does not hold for odd radices, even for arbitrarily large p_2, as shown by the following counterexample. With radix $\beta = 3$, for $p_1 = 4$, $p_2 \geq 4$, p_2 odd, for $x = 2001_3$ and $y = 2_3$*

$$\circ_1(\circ_2(x/y)) = \circ_1(\circ_2(1000.111\ldots_3)) = \circ_1(1000.1\ldots1_3) = 1000_3$$

which differs from $\circ_1(1000.111\ldots_3) = 1001_3$ when $\circ_1$ and $\circ_2$ are roundings to nearest with tie-break to even.

However, the result can be recovered by restricting the choice of tie-break rules to directed rules such as round to nearest with tie-break away.

THEOREM 32. *(**double_round_div_rna_FLX**) When β is odd, for $p_1, p_2 \in \mathbb{Z}$, if $p_2 \geq p_1$, then for $\circ_1^A$ and $\circ_2^A$ roundings to nearest with tie-break away, respectively in* FLX_{p_1} *and* FLX_{p_2}*:*

$$\forall x, y \in \mathrm{FLX}_{p_1}, y \neq 0 \Rightarrow \circ_1^A\big(\circ_2^A(x/y)\big) = \circ_1^A(x/y)\,.$$

Table I. Hypotheses under which double rounding is innocuous for the FLX format (β is the radix and p_1 and p_2 are the two precisions involved (c.f., Section 1 for formal definitions)). Division for odd radices is only valid with tie-break away.

	$\beta = 2$	even $\beta \geq 4$	odd β
$+, -$	$p_2 \geq 2p_1 + 1$	$p_2 \geq 2p_1$	$p_2 \geq p_1$
$\times$	$p_2 \geq 2p_1$	$p_2 \geq 2p_1$	$p_2 \geq p_1$
$/$	$p_2 \geq 2p_1$	$p_2 \geq 2p_1$	$p_2 \geq p_1$
$\sqrt{}$	$p_2 \geq 2p_1 + 2$	$p_2 \geq 2p_1 + 1$	$p_2 \geq p_1$

Table II. Hypotheses under which double rounding is innocuous for the FLT format, in addition to hypotheses of Table I (e_{min1} and e_{min2} are the two minimal exponents involved). Division for odd radices is only valid with tie-break away.

	$\beta = 2$	even $\beta \geq 4$	odd β
$+, -$	$e_{min2} \leq e_{min1}$	$e_{min2} \leq e_{min1}$	$e_{min2} \leq e_{min1}$
$\times$	$e_{min2} \leq 2e_{min1}$	$e_{min2} \leq 2e_{min1}$	$e_{min2} \leq e_{min1}$
$/$	$e_{min2} \leq e_{min1} - p_1 - 2$	$e_{min2} \leq e_{min1} - p_1 - 2$	$e_{min2} \leq e_{min1}$
$\sqrt{}$	$(e_{min2} \leq e_{min1} - p_1 - 2$	$(e_{min2} < e_{min1} - p_1$	$e_{min2} \leq e_{min1}$
	$\vee\, 2e_{min2} \leq e_{min1} - 4p_1 - 2)$	$\vee\, 2e_{min2} \leq e_{min1} - 4p_1)$	

Table III. Hypotheses under which double rounding is innocuous for the FTZ format, in addition to hypotheses of Table I. Division for odd radices is only valid with tie-break away.

	even β	odd β
$+, -$	$e_{min2} + p_2 \leq e_{min1} + 1$	$e_{min2} + p_2 \leq e_{min1} + p_1$
$\times$	$e_{min2} + p_2 \leq 2e_{min1} + p_1$	$e_{min2} + p_2 \leq e_{min1} + p_1$
$/$	$e_{min2} + p_2 < e_{min1}$	$e_{min2} + p_2 \leq e_{min1} + p_1$
$\sqrt{}$	$2\,(e_{min2} + p_2) \leq e_{min1} + p_1 \leq 1$	$e_{min2} + p_2 \leq e_{min1} + p_1$

Proof. According to Lemma 18, the result holds when x/y is not a midpoint. It also holds when it is, since the directed tie-break rule ensures that midpoints are always rounded in the same direction. □

6. CONCLUSION

Although all the results were illustrated throughout the paper on the FLX floating-point format (no underflows), they remain valid under mild conditions on the FLT (gradual underflow) and FTZ (abrupt underflow) formats with underflows. Indeed, all our Coq proofs are carried out at the level of $\mathbb{F}_\varphi$ generic formats and results for the FLX, FLT and FTZ formats with or without underflows are just corollaries immediatly derived from the former. The precise hypotheses under which double rounding has been proved innocuous are summarized in Tables I, II and III.

Once the proofs, originally expressed in terms of bit patterns [Fig95], have been translated to the more algebraic form seen throughout the paper, Coq proofs appeared rather easy to perform. A key factor in this easyness was the high genericity of the definitions in the Flocq library [BM11]. For instance, most of our proofs are done for *any* rounding to nearest, for any tie, by just using the definition of rounding to nearest available in Flocq. It should also be noticed that the library appeared pretty complete since no definition and only a few (13) lemmas were added to it.

All our Coq developments (6.9 kloc) are part of the Flocq library [BM11] which is available[1] under an open source license at http://flocq.gforge.inria.fr/.

Thus, we have not only formally proved previously-known results on innocuousness of double rounding of basic arithmetic operations [Fig95] but also extended them to radices other than two (such as radix 10 introduced in the last revision of the IEEE754 norm [IEE08]) and underflow cases (either gradual or abrupt). To the extent of author knowledge, although they belong to the folklore in the floating-point arithmetic community, no proof, neither pen and paper nor mechanical, can be found in the literature. In particular, it is now formally proved that double rounding of addition/subtraction, multiplication, division, and square root, is innocuous for IEEE754 binary32 format when using binary64 as intermediate precision. This could, for instance, be used in the verified C compiler Compcert [BJLM13] to perform constant folding while cross compiling for a binary32 target on a binary64 host. This may also be of interest for some processors, such as the one using the POWER (ancestor of PowerPC) instruction set, which have binary64 arithmetic instructions but lack binary32 ones. The RAD6000, a radiation-hardened board used onboard various spacecrafts, belongs to this category.

ACKNOWLEDGMENTS

The author wants to deeply thank Sylvie Boldo and Guillaume Melquiond for their help for this work and the anonymous reviewers for their comments.

References

[BCHPM04] Yves Bertot, Pierre Castéran, Gérard (informaticien) Huet, and Christine Paulin-Mohring. *Interactive theorem proving and program development : Coq'Art : the calculus of inductive constructions*. Texts in theoretical computer science. Springer, Berlin, New York, 2004. Données complémentaires http://coq.inria.fr.

[BJLM13] Sylvie Boldo, Jacques-Henri Jourdan, Xavier Leroy, and Guillaume Melquiond. A Formally-Verified C Compiler Supporting Floating-Point Arithmetic. In Alberto Nannarelli, Peter-Michael Seidel, and Ping Tak Peter Tang, editors, *IEEE Symposium on Computer Arithmetic*, pages 107–115. IEEE Computer Society, 2013.

[BM11] Sylvie Boldo and Guillaume Melquiond. Flocq: A Unified Library for Proving Floating-point Algorithms in Coq. In *Proceedings of the 20th IEEE Symposium on Computer Arithmetic*, pages 243–252, Tübingen, Germany, July 2011.

[Coq12] The Coq development team. *The Coq proof assistant reference manual*, 2012. Version 8.4.

[Fig95] Samuel A. Figueroa. When is Double Rounding Innocuous? *SIGNUM Newsl.*, 30(3):21–26, July 1995.

[IEE08] IEEE Computer Society. IEEE Standard for Floating-Point Arithmetic. *IEEE Standard 754-2008*, 2008.

[MDMM13] Érik Martin-Dorel, Guillaume Melquiond, and Jean-Michel Muller. Some issues related to double rounding. *BIT Numerical Mathematics*, 53(4):897–924, 2013.