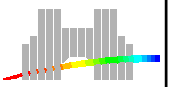
Marc Daumas
Senior scientist
French National Center for Scientific Research (CNRS)
LIP Computer science laboratory
http://www.ens-lyon.fr/~daumas/

Arenaire

Joint project



ENS at Lyon

CNRS

INRIA

Past and future formalizations of the IEEE 754, 854 and 754R standards

Introduction, motivations, survey
Achievements in the French AOC action
Next steps

Talk presented to the IEEE 754R committee
July 18th, 2002 at Cupertino, California
http://grouper.ieee.org/groups/754/

I – Introduction, motivations and survey

Two examples of failure

Identified groups and systems

Endeavour US Space Shuttle maiden flight

Software failure for the rendezvous maneuvers with the Intelsat satellite

– The numeric code was specified and tested by IBM but the bug is input dependent
– The code searches a double precision solution between two single precision bounds
– The bounds are loosely updated by rounding the double precision intermediate values
– When the code converges quickly, the bounds may exclude the exact result

The software was updated "live" from the earth

If you didn't test it, it doesn't work

Testing is not sufficient, formal validation is necessary

– Bug condition may be very uncommon
  • For example double precision rounding of the exponential with 105 bits of accuracy can be erroneous in 28 cases out of 18,014,398,509,481,984 (Muller *et al.*)
– Formal verification forces precise specification

If you didn't *validate* it, it doesn't work

Synchronizing clocks in the presence of faults

Algorithm published in 1985 by Lamport and Melliar Smith, used and studied worldwide

– Peer review for publication in the Journal of the ACM with a 3-year revision process
– Formal, theoretical and software developments spurring from related questions
  (PODC Influential Paper Award received in 2000 by Lamport for a 1978 related paper)
– Columbia US Space Shuttle maiden flight postponed by a clock synchronization failure

The proof of the algorithm contains errors but the algorithm is correct

– Four lemmas out of five and the main induction are wrong
– The errors were prompted in 1991 by an effort to check the proof on a automatic system

Human interactions (lectures, discussions and peer reviews)
might not be adapted to guarantee the technological developments of the future

– Production costs are reduced by reuse of existing components in small development teams
– Time to market constraints forbid extensive studies of industrial applications
– More and more applications may be safety critical and failure rates must drop drastically

References related to the two examples of failure

If you didn't test it, it doesn't work, B. Colwell
IEEE Computer, vol. 35, no. 5, pp. 11-13, 2002.

Computer related risks, P. G. Neumann
ACM Press, 1995.

Moyens arithmétiques pour un calcul fiable, V. Lefèvre, PhD thesis
École Normale Supérieure de Lyon, Lyon, France, 2000.

Synchronizing clocks in the presence of faults, L. Lamport and P. M. Melliar-Smith
Journal of the ACM, vol. 32, no. 1, pp. 52-78, 1985.

Formal verification of algorithms for critical systems, J. Rushby and F. von Henke
Proceedings of the Conference on Software for Critical Systems, pp. 1-15, 1991.

http://www.ens-lyon.fr/LIP/Arenaire/Bibliographies/daumas_ref.html.en
http://liinwww.ira.uka.de/bibliography/Math/computer.arithmetic.1.html
http://www.ens-lyon.fr/~daumas

Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

I – Introduction, motivations and survey

Two examples of failure

Identified groups and systems

Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Logic tools for proof development and checking

ACL2, R. Boyer, M. Kaufmann and J. Moore, 1979

– First order quantifier free logic
– Fully automatic and goal directed proofs with natural deduction (tactics)

HOL (Higher Order Logic), M. Gordon, 1988

– Church's theory of simple types (Curry Howard isomorphism)
– Small trusted logical core with higher order inferences decomposed to primitives

PVS (Prototype Verification System), S. Owre, and N. Shankar, 1992

– Strongly typed higher order logic with dependent types and predicate subtypes
– Automatic decision and BDD procedures integrated with the type checker
– Backward proof only using sequent representation, no separate engine, five tactics

Coq, T. Coquand, 1987

– Strongly typed higher order logic with dependent types
– Very small inference engine (300-400 lines of code)

Hardware oriented floating point specifications

Z, G. Barrett, 1989

– Oxford University and INMOS IMS T800 Transputer
– Formal specification of the IEEE standard for binary floating point arithmetic in hardware
– Unpack the operands, perform the operation, round and pack the result

ACL2, M. Kaufmann, T. Lynch, J. Moore and D. Russinoff, 1998

– University of Texas and Advanced Micro Devices (K5 – K7)
– Probably the most achieved RTL (Register Transfer Level) specification on bit vectors
– Limiting use of first order logic and built-in rational arithmetic

PVS, C. Berg, C. Jacobi and W. Paul, 1995

– Saarland University, VAMP verified architecture microprocessor
– Automatic translation tool of the PVS specification to Verilog
– Developments specific to IEEE rounding (guard, round and sticky)

Fundamental floating point specifications

PVS, P. Miner and **HOL,** V. Carreño, 1995

– NASA Langley Research Center probably on avionics application
– Radix independent IEEE 854 and binary IEEE 754 digit oriented specifications
– Informal construction of tables to summarize all the possible cases

HOL, J. Harrison, 1997

– University of Cambridge, now Intel
– Generic definitions specialized to particular formats with medium level lemmas
– Direct translation of the standard specially for the rounding function
– Polynomial approximation of the exponential function (Chebychev, Sturm)

Coq, S. Boldo, M. Daumas, L. Rideau and L. Théry, 2001

– INRIA funded collaboration of proof checking and computer arithmetic scientists
– Academic development freely available and maintained on the internet
– Higher order logic allows better translation of the intentions specially for the exceptions

II – Achievements in the French AOC action

Abstract studies
Two's complement (Texas Instrument)
Expansions of floating point numbers

The floating point round off error can be represented

Question raised by G. Bohlender, W. Walter, P. Kornerup and D. Matula

- Promote hardware functions to extend the available precision in floating point arithmetic
- Every operation may return an exact correcting term

Addition	$x + y - x \oplus y$	Construction by W. Kahan, O. Møler
Multiplication	$x \times y - x \otimes y$	Construction by T. Dekker
Division	$x - y \times \mathrm{fl}(x / y)$	$y \neq 0$
Square root	$x - (\mathrm{RN}(\sqrt{x}))^2$	$x \geq 0$, only possible when rounding to the nearest

Examine the possible underflows and provide a tight necessary and sufficient condition

Addition	The rounding mode must be set to the nearest
Multiplication	(n_x, e_x) and (n_y, e_y) bounded pairs, mapped to x and y with $e_x + e_y \geq -e_{MIN}$
Division	(n_y, e_y) and (n_q, e_q) bounded, mapped to y and $\mathrm{fl}(x / y)$ with $e_y + e_q \geq -e_{MIN}$ and $\lvert q \rvert \neq \lambda$ or $\lvert x/y \rvert \geq \lambda/2$ with λ smallest positive number
Square root	(n_q, e_q) bounded, mapped to $\mathrm{RN}(\sqrt{x})$ with $2e_q \geq -e_{MIN}$

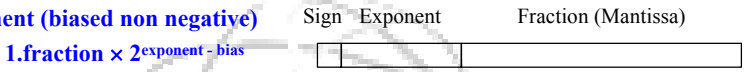
Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

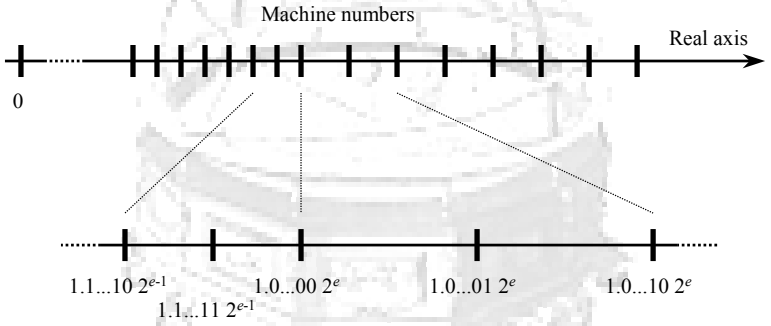
IEEE standard floating point numbers

Binary fraction (implicit 1)

Binary exponent (biased non negative)

$x = (-1)^{sign} \times 1.fraction \times 2^{exponent - bias}$





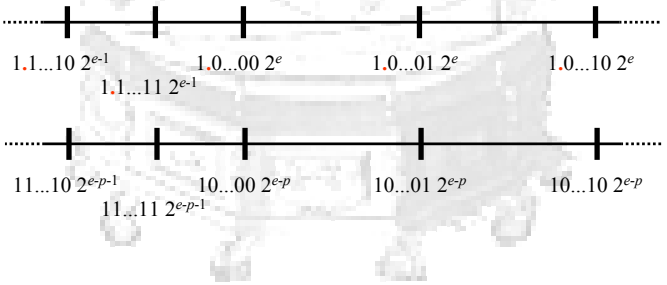
Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Formalized floating point numbers

Floating point pair of signed integers — **(mantissa, amplitude)**

Mapping function to the real numbers — $x = \text{mantissa} \times b^{\text{amplitude}}$

Bounded pair — $|\text{mantissa}| < m_{MAX}$ and amplitude $\geq -e_{MIN}$



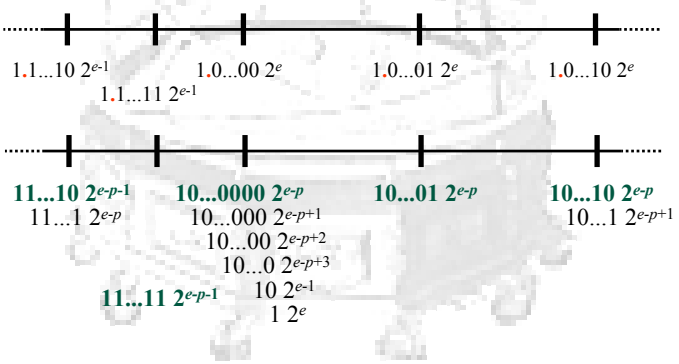
Normal floating point numbers

Floating point pair of signed integers — **(mantissa, amplitude)**

Mapping function to the real numbers — $x = \text{mantissa} \times b^{\text{amplitude}}$

Bounded pair — $|\text{mantissa}| < m_{MAX}$ and amplitude $\geq -e_{MIN}$

Normal pair — **Bounded and $b \times |\text{mantissa}| \geq m_{MAX}$**



Subnormal numbers

Example with the "toy" system

Radix 10 notation

3 digit mantissa

- Between 0 and 999
- Normal mantissa between 100 and 999

Amplitude

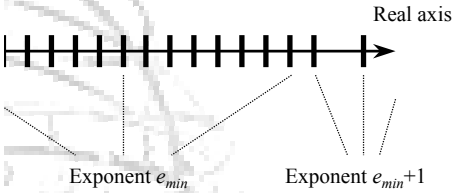
- Between -9 and +10

First positive machine numbers

100×10^{-9}, 101×10^{-9}, 102×10^{-9} …

then 999×10^{-9}, 100×10^{-8}, 101×10^{-8} …

tion close to 0

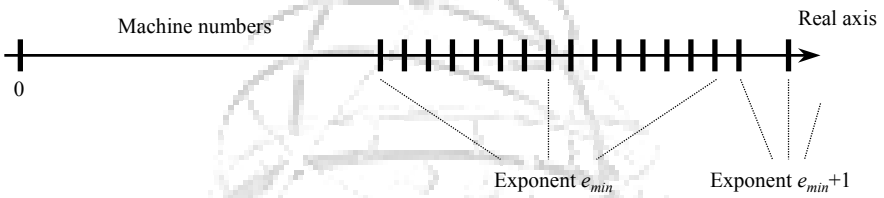


rmal numbers (TINY)

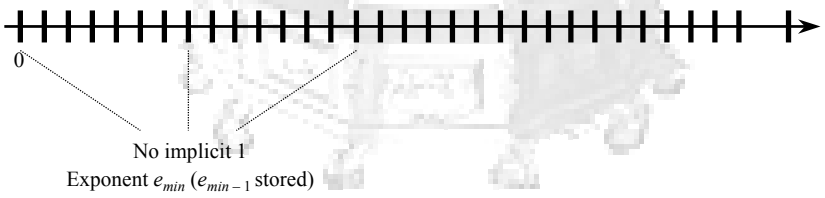
Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Subnormal numbers

Normal notation close to 0



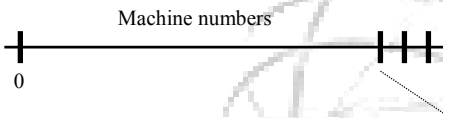
Notation with subnormal numbers (TINY)



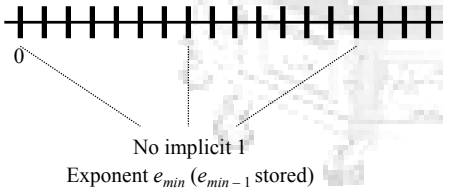
Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Subnormal numbers

Normal notati



Notation with subnorı



Example with the "toy" system

Radix 10 notation

3 digit mantissa

- Between 0 and 999
- Normal mantissa between 100 and 999

Amplitude

- Between -9 and +10

Examples

142×10^{5} or -999×10^{-9} …

15×10^{-9} or 0 …

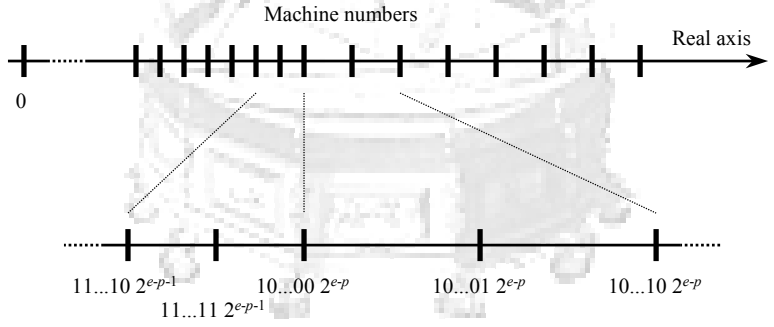
Subnormal pairs are native to the data type

Discard subnormals to Flush to Zero

Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

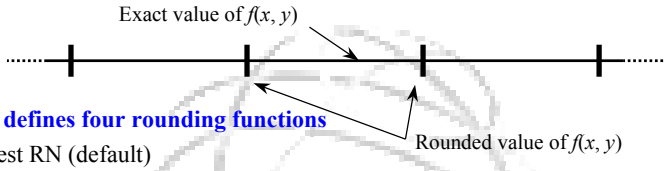
Formalize a redundant floating point number system

Floating point pair of signed integers	**(mantissa, amplitude)**
Mapping function to the real numbers	**x = mantissa $\times$ b$^{\text{amplitude}}$**
Bounded pair	**$\lvert\text{mantissa}\rvert < m_{MAX}$ and amplitude $\geq -e_{MIN}$**
Normal pair	**bounded and $b \times \lvert\text{mantissa}\rvert \geq m_{MAX}$**
Subnormal pair	**amplitude = $-e_{MIN}$ and $b \times \lvert\text{mantissa}\rvert < m_{MAX}$**
Canonic pair	**either normal or subnormal**



Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Rounding modes



The standard defines four rounding functions

– To the nearest RN (default)
– Truncated (towards 0) RZ
– Directed (up RU or down RD)

Tie breaking rule

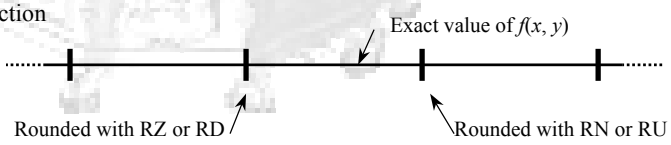
– Odd and even mantissas alternate for any radix
– Most our proofs are independent of the tie breaking rule

Axiomatic definition of the rounding relations

– Function would not allow redundant results
– Compatible and total subset of $\mathbf{F} \times \mathbf{R}$
– The relation is monotonous (non decreasing)
– The result is the rounded up or down value
  • Future evolution to a projection

Higher order logic allow quantification on

– A generic rounding mode
– Any rounding mode to the nearest pair



Benefits and surprises of the complete approach

Expressing a necessary and sufficient condition for the underflow to be harmful

– Quantification on the redundant notations of the floating point system
– Replaces the notion of least significant non zero digit

Division

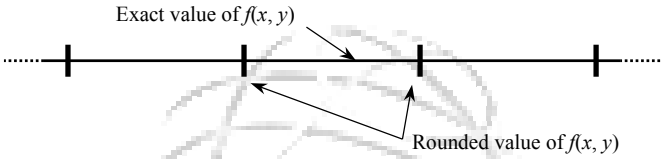
– The condition $|q| \neq \lambda$ or $|x/y| \geq \lambda/2$ is always true when rounding to the nearest
– The "remainder" $x - y \times \mathrm{fl}(x / y)$ can be much larger in magnitude than the dividend
– Example on a toy system rounding up with $-e_{MIN} = -9$ when $x = 10^{-4}$ and $y = 10^{9}$
  $x / y = 10^{-13}$ $\quad \mathrm{fl}(x / y) = 10^{-9}$ $\quad x - y \times \mathrm{fl}(x / y) = 10^{-4} - 1 = -0.9999$

Square root

– On a strange floating point system, the square root of a number can be subnormal
– Example on a five digit mantissa system with $-e_{MIN} = -3$
  $x = 1$ $\quad \sqrt{x} = \mathrm{RN}(\sqrt{x}) = 1$ is subnormal

Further developments on Texas Instrument TMS 320 C3x digital signal processing circuit

Implementation of the arithmetic operators



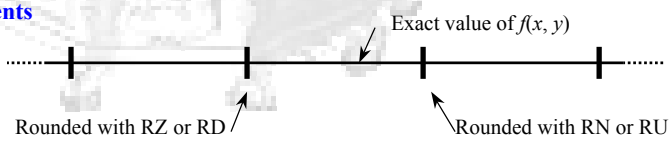
Exact operations on the floating point pairs

– Compatibly defined on the floating point pairs
– Addition, subtraction and multiplication
  • Both exact and rounded pairs
  • User specified rounding mode
  • Quantified on any generic rounding mode

Algebraic operators on the real values

– Defined on the real mapping of the input
– Division and square root
  • Only rounded pairs
– Exact remainder (new proof)
  Replacing a proof based on the algorithm

Transcendental functions available in Coq for future developments



Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

II – Achievements in the French AOC action

Abstract studies

Two's complement (Texas Instrument)

Expansions of floating point numbers

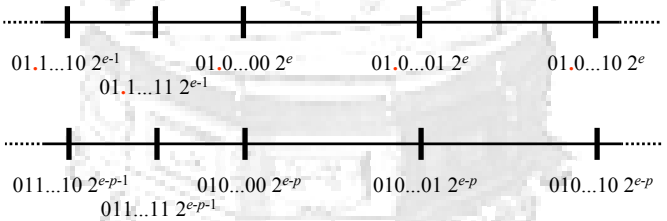
Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Two's complement formalized floating point numbers

Floating point pair of signed integers **(mantissa, amplitude)**

Mapping function to the real numbers $x = \text{mantissa} \times b^{\text{amplitude}}$

Bounded pair $-m_{MIN} < \text{mantissa} < m_{MAX}$ and amplitude $\geq -e_{MIN}$



Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Revisiting Sterbenz's theorem with Coq

If $\frac{1}{2}y \leq x \leq 2y$ then $\text{fl}(x - y) = x - y$

Radix independent for IEEE-like implementations

Sterbenz's theorem is a property of the number system: $x - y$ has to be represented

– Generic theorem: if $y \leq x \leq 2y$ then $x - y$ can be represented

– Sterbenz's theorem follows if any number can be negated

A number system where $m_{MAX} \neq m_{MIN}$ is stable by negation if and only if $|m_{MAX} - m_{MIN}| = 1$ and the radix divides $\max(m_{MAX}, m_{MIN})$

Rounding the extended floating point intermediate result, $x - y$ is found on the TMS 320

Texas Instrument TMS 320 C3x could be a first step to a good implementation

– Two's complement mantissa defines almost an IEEE-like floating point system

– Negative numbers use a very different storage compare to an IEEE-like implementation

– The system can be extended to handle subnormal numbers, precise rounding and exceptions

Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Some algebra laws disappear

The multiplication should be associative

– $(151 \otimes 145) \otimes 145 = 219 \times 10^2 \otimes 145 \times 10^0 = 318 \times 10^4$

– $151 \otimes (145 \otimes 145) = 151 \times 10^0 \otimes 210 \times 10^2 = 317 \times 10^4$

Introduce limited relative error

The addition should be associative

– $(100 \times 10^{-3} \oplus 101) \oplus (-100) = 101 \oplus (-100) = 1$

– $100 \times 10^{-3} \oplus (101 \oplus (-100)) = 100 \times 10^{-3} \oplus 100 \times 10^{-2} = 1{,}1$

Possibly introduce a huge relative error – Catastrophic cancellation

The multiplication should distribute over the addition (with or without cancellation)

– $(102 \oplus 349) \otimes 103 = 451 \otimes 103 = 465 \times 10^2$

– $102 \otimes 103 \oplus 349 \otimes 103 = 105 \times 10^2 \oplus 359 \times 10^2 = 464 \times 10^2$

– $(105 \oplus -104) \otimes 349 = 1 \otimes 349 = 349$

– $(105 \oplus -104) \otimes 349 = 366 \times 10^2 \oplus -363 \times 10^2 = 300$

More properties disappear

Cauchy's inequality

$$||u||\,||v|| \geq |u.v|$$

Not necessarily true

Standard deviation formula

$$(\Sigma_k x_k^2 - (\Sigma_k x_k)^2) / (n\,(n-1))$$

Eventually negative

Analytic geometry becomes difficult

Orientation of a triangle

$$(x_B - x_A)(y_C - y_A) - (x_C - x_A)(y_B - y_A)$$

Positive if and only if ABC is direct

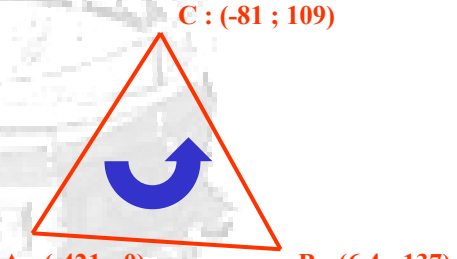
$(6{,}4 + 421) \times (109 + 0) + -(-81 + 421) \otimes (137 + 0) = 427{,}4 \times 109 - 340 \times 137 = 6{,}6$

$(6{,}4 \oplus 421) \otimes (109 \oplus 0) \oplus -(-81 \oplus 421) \otimes (137 \oplus 0)$

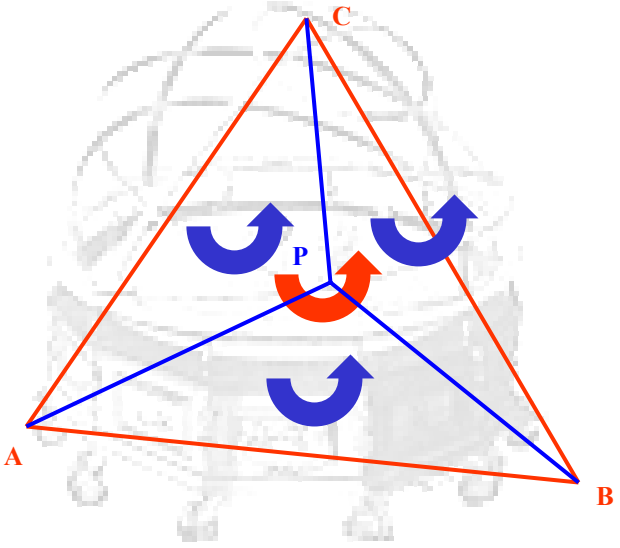
$= 427 \otimes 109 \oplus -340 \otimes 137$

$= 465 \times 10^2 \oplus -466 \times 10^2$

$= -100$



Geometry is no longer Euclidian



Some mathematic rules remain

The addition and the multiplication are commutative

Existence of an opposite that is the negation

$x \oplus (-x) = 0$

Uniquely defined if the system treats subnormal numbers

0 and 1 respectively zero and unity of the addition and the multiplication

Weak compatibility of the order relations with the arithmetic operators

$u < v$ implies $u \oplus w \leq v \oplus w$

$u < v$ et $0 < w$ implies $u \otimes w \leq v \otimes w$

Norm of a vector when the rounding mode is set to the nearest or up

$-1 \leq x_i / \sqrt{\Sigma x_j^2} \leq 1$

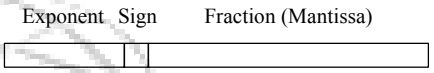
Two's complement floating point numbers

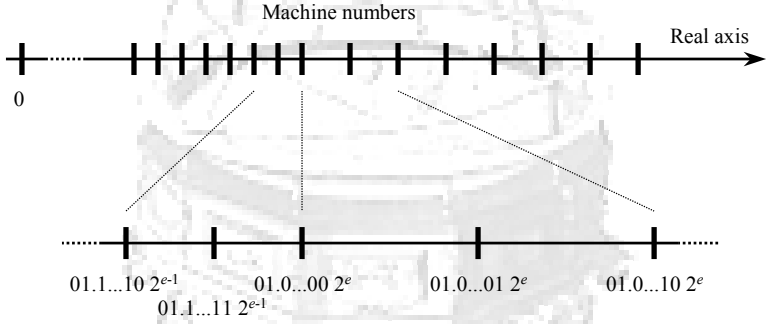
Binary fraction (implicit 1)

Binary exponent (biased non negative)

Two complement mantissa

$x = (S(1-S).\text{fraction})_{2C} \times 2^{\text{exponent - bias}}$





Implemented on Texas Instrument TMS/SMJ 320 C3x digital signal processing circuit

II – Achievements in the French AOC action

Abstract studies

Two's complement (Texas Instrument)

Expansions of floating point numbers

Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Multiple precision library: floating point expansions

Expansion **Non adjacent expansion**

The components are floating point machine numbers, the value is the sum $\mathbf{x} = x_{n-1} + \ldots + x_1 + x_0$

The extracted series of non zero components is sorted by magnitude

Non overlapping

(There is at least one zero bit between two non zero bits of neighboring components)

Addition, multiplication and division operators implemented

Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Benefits of the automatic proof checking

A proof/algorithm can be trusted as soon as it is checked by the computer

– Writing a computer validated proof takes much longer than writing a pen and paper proof

– Trusting a pen and paper proof requires *many long* discussions with careful reviewers

The early developments of our expansion library were limited by the trust in our proofs

– Many foreseeable questions were not explored to maintain the proof length to a length appropriate for human review

– Some choices were left to the software by implementing tests although one of the branches was very improbable

As the foundation properties of our library were checked

– More questions were explored using automatic proof checking

– The algorithms were changed as

- We failed in proving that the improbable branches were impossible
- We proved that the modified algorithm is better than the first one

Behavior of Malcolm's algorithm proved in Coq

Real RAM machine

```
A := 1; B := 1;
while (A + 1 – (A + 1)) = 0
    do A := A × 2;
while (A + B – (A + B)) ≠ 0
    do B := B + 1;
write B;
```

The real RAM program

– Never leaves the first loop

– Never enters the second loop

Floating point machine

```
A := 1; B := 1;
while (A ⊕ 1 ⊕ -(A ⊕ 1)) = 0
    do A := A ⊗ 2;
while (A ⊕ B ⊕ -(A ⊕ B)) ≠ 0
    do B := B ⊕ 1;
write B;
```

The implemented program writes the floating point radix

Exact sum operator specified and proved in Coq

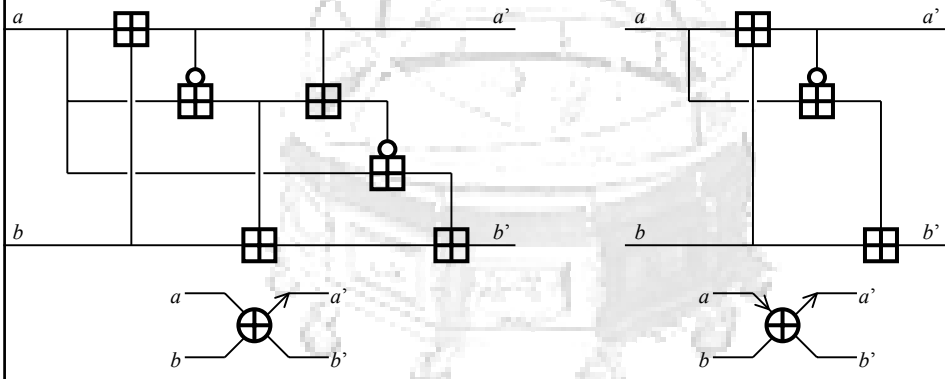
Exact sum $a' + b' = a + b$

– a' = RN $(a + b)$

Conditional sum

– New validated sufficient condition

– Same property on a' and b'

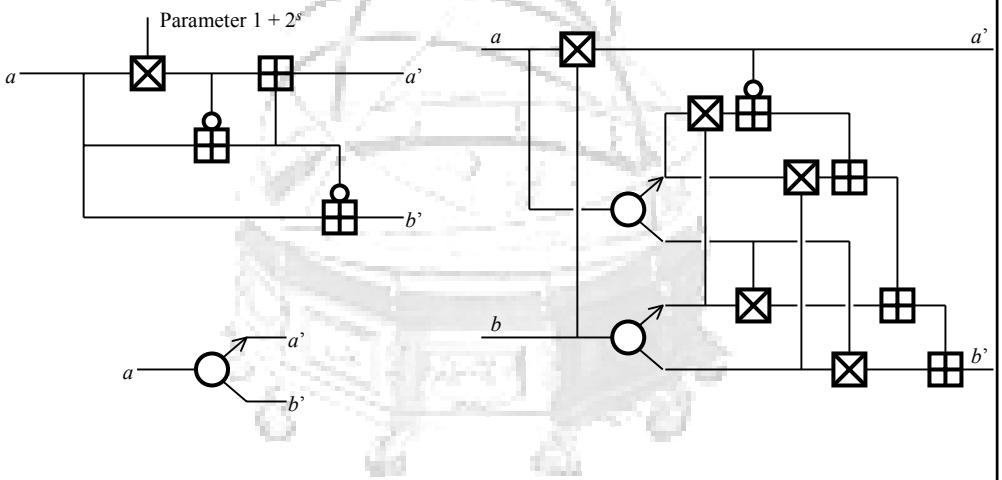


Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Exact product operator specified but not proved in Coq

Split a number in two half-precision numbers

Accumulate the partial products



Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

References on work in the French AOC action

A generic library of floating-point numbers and its application to exact computing

http://www.ens-lyon.fr/~daumas/SoftArith/DauRidThe01.ps

14th Int. Conf. on Theorem Proving in Higher Order Logics, pp. 169-184, 2001.

A mechanically validated technique for extending the available precision

http://www.ens-lyon.fr/~daumas/SoftArith/BolDau01b.pdf

35th IEEE Asilomar Conf. on Signals, Systems, and Computers, pp. 1299-1303, 2001.

Properties of the subtraction valid for any floating point system

ftp://ftp.inria.fr/INRIA/publication/publi-ps-gz/RR/RR-4473.ps.gz

ERCIM 7th Int. Workshop on Formal Methods for Industrial Critical Systems, 2002.

Computer validated proofs of a toolset for adaptable arithmetic

http://arxiv.org/pdf/cs/0107025

Submitted to the Journal of the ACM.

http://www.ens-lyon.fr/~sboldo/coq/

http://www-sop.inria.fr/lemme/AOC/

III – Next steps

Concluding on work in the French AOC action

Exceptions and non numeric values

Annotated revised IEEE 754 standard

Standard compliance discussion group

Quick and dirty *versus* neat and too late

The road of validated proof is still long

- Extensions to existing work (real numbers…) 14 definitions and 204 theorems
- Definition of floating point arithmetic 60 definitions and 419 theorems
- Properties of floating point arithmetic 142 theorems
- Proof of program correctness 6 definitions and 84 theorems

Not only state and solve difficult properties and inductions but also…

… Develop tools, libraries and interfaces

… Reach for the huge common mathematical knowledge: Series, polynomials, linear algebra, function analysis…

Evaluate and control hardware / software / system dependability

- Safety critical systems (flight control computer…)
- Embedded and hidden computers (car…)
- Health, law enforcement, day-life…

Basic definitions

A very generic definition of the floating point pair

```
Record float : Set := Float {Fnum: Z; Fexp: Z }.
Definition FtoR := [x : float] (Rmult (Fnum x) (powerRZ (IZR radix) (Fexp x))).
```

Introduce the notion of bounds and the special canonic pairs

```
Record Fbound : Set := Bound {vNum: positive; dExp: nat }.
Definition Fbounded := [b : Fbound] [d : float]
    (Zlt (Zabs (Fnum d)) (POS (vNum b))) /\ (Zle (Zopp (dExp b)) (Fexp d)).
Definition Fnormal := [p : float]
    (Fbounded b p) /\ (Zle (POS (vNum b)) (Zabs (Zmult radix (Fnum p)))).
Definition Fsubnormal := [p : float]
    (Fbounded b p) /\ ((Fexp p) = (Zopp (dExp b)) /\
    (Zlt (Zabs (Zmult radix (Fnum p))) (POS (vNum b)))).
Definition Fcanonic := [a : float] (Fnormal a) \/ (Fsubnormal a).
```

More definitions

Define functions on canonic floating point pairs

```
Definition Fnormalize := [p : float]
  Cases (Z_zerop (Fnum p)) of (left _) => (Float ZERO (Zopp (dExp b)) | (right _) =>
    (Fshift radix (min (minus precision (Fdigit radix p)) (absolu (Zplus (dExp b) (Fexp p))))
    p) end.
Definition Fulp := [p : float] (powerRZ radix (Fexp (Fnormalize radix b precision p))).
```

Properties of the rounding relations

```
Definition TotalP := [P : R -> float -> Prop] (r : R) (Ex [p : float] (P r p)).
Definition CompatibleP := [P : R -> float -> Prop]
  (r1, r2 : R) (p, q : float) (P r1 p) -> r1 == r2 -> <R> p == q -> (Fbounded b q) -> (P r2 q).
Definition MinOrMaxP := [P : R -> float -> Prop]
  (r : R) (p : float) (P r p) -> (isMin b radix r p) \/ (isMax b radix r p).
Definition MonotoneP := [P : R -> float -> Prop]
  (p, q : R) (p', q' : float) (Rlt p q) -> (P p p') -> (P q q') -> (Rle p' q').
```


The defined rounding modes

```
Definition RoundedModeP := [P : R -> float -> Prop]
   (TotalP P) /\ ((CompatibleP P) /\ ((MinOrMaxP P) /\ (MonotoneP radix P))).
Definition isMin := [r : R] [min : float] (Fbounded b min) /\
   ((Rle min r) /\ ((f : float) (Fbounded b f) -> (Rle f r) -> (Rle f min))).
Definition isMax := [r : R] [max : float] (Fbounded b max) /\
   ((Rle r max) /\ ((f : float) (Fbounded b f) -> (Rle r f) -> (Rle max f))).
Definition ToZeroP := [r : R] [p : float]
   (Rle R0 r) /\ (isMin b radix r p) \/ ((Rle r R0) /\ (isMax b radix r p)).
Definition ToInfinityP := [r : R] [p : float]
   (Rle r R0) /\ (isMin b radix r p) \/ ((Rle R0 r) /\ (isMax b radix r p)).
Definition Closest := [r : R] [p : float] (Fbounded b p) /\
   ((f : float) (Fbounded b f) -> (Rle (Rabsolu (Rminus p r)) (Rabsolu (Rminus f r)))).
Definition EvenClosest := [r : R] [p : float] (Closest r p) /\
   ((FNeven b radix precision p)  \/ ((q : float) (Closest r q) -> <R> q == p)).
```

III – Next steps

Concluding on the work of the French AOC action

Exceptions and non numeric values

Annotated revised IEEE 754 standard

Standard compliance discussion group

Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Exception definition and handling in Coq

Use the expressiveness of the logic tool to define the exceptional situations

– Define two underflow exceptional situations (tiny and possibly inexact)
  - Before (exact operation) or after rounding (extended domain for the amplitude)
  - Consistent choice for all the operations
  - Not necessarily across all the implemented data types
– Invalid operations that define the signaling and quiet Not a Numbers (NaN)
– Overflow situation based on the rounding mode
– Division by zero with the signed infinities and the negative zero (non numeric values)
– Inexact result exception

Relation between the inputs, the exact operation and the rounding mode $F^2 \times Op \times Rnd$

Prove algorithms with an upper bound on the amplitude

– State necessary and sufficient conditions for the result to be faithful (overflow free)
– Use an intermediate upper bound on the amplitude such as IA64 internal format for the division

Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Status registers

Tests in the program modify conditions codes in the processor state

Program history must handle straightforward behavior…

– Maintain two separate fields in the processor state
  • Exception flags of the last operation (current)
  • Sticky flags of all the operations since flag reset (accrued)
– Generate the current exception flags and update the accrued flags if no handler is activated

…as well as Intel implementations

– Maintain only one unique field for the sticky flag
– Update the sticky flags prior to handler activation
– Activate any handler based on the sticky flags and the interruption mask
– Example: An addition may provoke a SIGFPE interruption motivate by a division by zero

Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

III – Next steps

Concluding on the work of the French AOC action
Exceptions and non numeric values
Annotated revised IEEE 754 standard
Standard compliance discussion group

Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Shortcomings of automatic proof checking

No formal specification built prior to the Pentium Pro bug (DAN-0411, 1997) defines the correct behavior of the floating point to integer conversion

A theorem is a mathematical object linking premises to conclusions via a proof

– An error may appear from a logical inconsistency or a bad behavior of the developing tools
– The specification of the floating point unit may have been poorly transcribed
– Some premises or conclusions of the theorems may be misunderstood
– Some part of the unit may not be validated

Build generic definitions, explore logical consequences and specialize the developments

Coding a grand unified theory of floating point arithmetic is impossible

– The differences in the underlying logic expressiveness cannot be crossed
– Building an automatic translator from one logical system to another would be long
– The author would have to prove its correct behavior

Annotate the standard from experiences on the development with the different logic tools

Annotated revised IEEE 754 standard

As a grand unified theory of floating point arithmetic will remain impractical

– We will prepare for the summer of 2003 a first draft of an annotated standard
– Based on our experience specifying floating point arithmetic with Coq
– With people from both
  • The computer arithmetic community
  • The automatic proof checking community
– Backed up by the growing set of proved facts that we have developed so far

We want to get other groups involved

– First work with people looking at fundamental aspects (Harrison at Intel, NASA)
– Start meeting with hardware oriented groups (Saarland, AMD) will also allow to
  • Integrate experiences from their developments
  • Use the annotated specification to define a hardware oriented specification
  • Prove correctness of the specification in regard to our fundamental developments

III – Next steps

Concluding on the work of the French AOC action

Exceptions and non numeric values

Annotated revised IEEE 754 standard

Standard compliance discussion group

Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA

Create an authority to discuss on standard compliance

For hardware and software implementations

– The success of testing software has proved the need for such an effort

– A discussion group could promote the development of existing and new testing tools

For specifications

– As incompatibilities will remain in natural languages and between the different formal tools

– Use natural languages and human comments to bridge the gap correctly

Such a committee should certainly NOT certify an implementation
but it could emphasize on a level of quality regarding

– The specification of the different problems

– Appropriate formal proof effort has been addressed

– Sufficient and discriminating tests have been performed or scheduled

– Documentation is accurate, sufficient and faithful

Marc Daumas – CNRS – LIP
ARENAIRE Project – ENS at Lyon – CNRS – INRIA