LEIBNIZ-RECHENZENTRUM
DER BAYERISCHEN AKADEMIE DER WISSENSCHAFTEN
8 MÜNCHEN 2 BARER STRASSE 21

München, den
Telefon-Nr. 21 05/393
FS-Nr. 05/24634
13 August 1976

Dear Colleague:

For your records I enclose a copy of Jeremy Du Croz' summary of the discussion on desirable features of a machine arithmetic, Wednesday, 23 June 1976 at Oak Brook. I plan to prepare a more elaborate report which could then be endorsed by the Working Group. To this end I extend the concluding remark of these minutes to those who were not present and cordially ask everybody to let me have his comments, opinions and ideas on pertinent points.

Yours sincerely,

[illegible]

Christian H. Reinsch

Distr.

| | | |
|---|---|---|
| E L Battiste | J Cody | I. Dahlstrand |
| T J Dekker | B Einarsson | B Ford |
| L D Fosdick | N Gastinel | C W Gear |
| P W Hemker | T E Hull | W M Kahan |
| C L Lawson | J T C Pool | J K Reid |
| J L Schonfelder | B T Smith | H J Stetter |

Summary of a discussion on

Machine Arithmetic

held on Wednesday 23rd June 1976

during the Workshop on Portability of Numerical Software

at Oakbrook, Illinois

| | | |
|---|---|---|
| present: | C. Reinsch | Leibniz-Rechenzentrum (Chairman) |
| | I. Dahlstrand | Lund University |
| | T. Dekker | University of Amsterdam |
| | J.J. Du Croz | N.A.G. Central Office |
| | P.W. Hemker | Stichting Mathematisch Centrum |
| | T.E. Hull | University of Toronto |
| | W.M. Kahan | University of California |
| | J.L. Schonfelder | University of Birmingham |
| | H.J. Stetter | Technische Hochschule, Wien. |

The following list of suggested topics for discussion was drawn up in advance by the chairman:

Concerning hardware:

1. Floating-point numbers and their representation in the machine.
2. Basic arithmetic operations.
3. Arithmetic comparisons.
4. Overflow, underflow, exception-handling.
5. Fixed-point arithmetic.
6. Documentation.

Concerning software and systems:

7. Radix conversion and fixed-floating conversion.
8. Elementary functions (e.g. $\sqrt{x}$ , $y^x$)
9. Ordering of arithmetic operations by compilers.
10. Precision control.
11. Special modes of arithmetic (e.g. interval, significance).

(Only topics 1. to 3. were discussed during the time available.)

The following approach was adopted on each topic:

either to specify a definite requirement to be met by the machine arithmetic, or to allot gradings to various possible types of operation (e.g. grade A would be the most preferred type, grade B would be acceptable).

## 1. The Floating-Point Number System

1.1. What base is preferred?

Arguments:

a) from a numerical point of view, base 2 is best, on grounds of economy of storage (especially with implicit first bit), and of minimizing errors and preserving algebraic relations during computation; but it has only a slight advantage over base 4.

b) however human beings use base 10, so the social advantages of using base 10 are overwhelming.

c) base 10 would abolish the need for base-conversion on input and output

d) little extra storage is needed to hold base 10 numbers if they are suitably coded (e.g. 3 decimal digits in 10 bits).

e) problems of wobbling precision in base 10 can be handled by extra precision in arithmetic registers.

f) manufacturers believe that base 10 arithmetic is expensive, so they are likely to continue to produce base 2 machines; however the widespread use of base 10 in pocket calculators is perhaps evidence that this is not a solid argument.

g) base 2 is more robust in the face of bad design; base 10 needs more care.

1.1. What base is preferred?

Arguments (contd)

h) there are no good arguments in favour of base 8 or base 16.

Conclusions:

|  |  |
|---|---|
| base 10 | grade A |
| base 2 | grade A- |
| base 4 | grade B. |

1.2. Is there a preferred representation of negative numbers?

Conclusions:

the representation of negative numbers (e.g. sign-magnitude, 1's-complement or 2's-complement) should make absolutely no difference to the arithmetic operations; therefore there must be no detectable anomalies such as asymmetic rounding or asymmetry in the range. However there was a preference for a sign-magnitude representation, because this makes it easier for a human being to interpret the machine-representation.

1.3. Is there a preferred range for the exponent?

Arguments:

a) It depends partly on the ease of response to overflow or underflow; if the range is very small, overflow and underflow will occur often.

b) If the range is too large, a program containing an error may continue to compute for a long time before causing overflow or underflow, by which time all trace of the original error may have been lost.

Conclusions:

The range should be wide enough for $\varepsilon^{\pm N}$ to be representable where $\varepsilon$ is the relative machine precision, and 10 is perhaps a suitable value for N, 4 being too small and 40 too large. The relation between the precision and the preferred range is neither a simple one nor a strong one.

1.4. Should the range be symmetric about 1?
(in the sense that if x is representable, then so also is 1/x).

Arguments:

a) Marked asymmetry (such as occurs in some CDC and Burroughs machines) is a nuisance, but not a disaster.

b) The asymmetry which is usually found on a binary machine with fractional normalization is negligible.

1.4. Should the range be symmetric about 1?

Arguments (contd)

c) Asymmetry could sometimes be reduced by normalizing the mantissa so that the point comes after the first digit (but this depends on the precise range of values of the exponent).

Conclusion:

A symmetric range is desirable, but not essential.

1.5. Should the single-precision numbers be a subset of the double-precision numbers?
(and should the representation of double precision numbers contain a single precision number as its leading part?)

Argument:

A major use for double precision quantities is to compute with them in combination with single precision quantities, so efficient conversion between the two is desirable.

Conclusion:

The essential requirement is to have efficient functions:

| | | |
|---|---|---|
| Shorten | - | to convert a double precision value to single precision |
| Long | - | to convert a single precision value to double precision |

such that the following relation always holds:

x = shorten (long(x)).

1.6. Should the exponent-range of double precision numbers be the same as that of single precision numbers?

Arguments:

a) It is sometime useful to have a wider range available.

b) If the range of double precision numbers is wider, overflow may occur on converting to single precision; but this does not matter if overflow is handled efficiently.

c) There might be problems in allowing for a larger exponent field in the stored representation of double precision numbers, especially if numbers have to be decoded between store and the arithmetic unit.

Conclusion:

The exponent range of double precision numbers must be at least equal to that of single precision numbers; it may be larger.

1.7. Should unnormalized numbers be allowed?

Arguments:

a) The usual relative error analysis breaks down if unnormalized numbers may occur.

b) The question is related to the handling of "partial underflow", which may occur, for example, in forming the difference of two small positive numbers: the unnormalized result may be exactly representable, but normalization would cause underflow. Which is preferable: to retain the unnormalized result or to indicate underflow? The information contained in the unnormalized result may be useful.

c) Grau's representation of floating-point numbers (as used in the Electrologica X8 and the Burroughs 5500) allows numbers to be represented which are not normalized in the usual sense. But it was not clear what advantages this representation offered, especially in view of (a).

Conclusion:

"Non-exceptional" numbers must always be normalized. Any "exceptional" result (such as overflow, underflow, partial underflow) must cause an interrupt, close to the point at which it is generated, giving the user options on how to handle it (to be considered under topic 4); exceptional numbers must not be allowed to appear and disappear without the user being made aware of them.

1.8. Representation of "exceptional numbers".

Conclusion:

It is desirable to reserve certain bit-patterns to represent, for example, an overflowed or underflowed result or an undefined variable. This can be done by reserving for this purpose one or two values of the exponent, or by taking advantage of the slight redundancy in coding a decimal representation into a bit pattern (see 1.1., argument (d)). Moreover the results of all possible arithmetic operations and comparisons involving such bit patterns must be specified, e.g. in a table.

## 2. Basic Arithmetic Operations

Arguments:

a) Ideally the result of any basic operation (+,-,*,/) should be correctly rounded to the nearest representable number, with a specific rule for the ambiguous case when there are two nearest such numbers; this rule must be symmetric about 0 and, preferably, locally unbiassed.

b) Faithful rounding (in which the result is always one of the two representable numbers on either side of the exact result) is also tolerable, provided that it is locally unbiassed.

c) Chopping, which is faithful but biassed, has been proven to be inferior in practical experiments.

## 2. Basic Arithmetic Operations (contd)

Conclusion:

| | |
|---|---|
| Correct rounding with locally unbiassed rule for ambiguous case | grade A+ |
| Correct rounding (with some other specifc rule for ambiguous case) | grade A |
| Faithful, and locally unbiassed, rounding | grade A- |

Also the results of negation and taking the absolute value of a representable number must always be exact.

Finally, the operations must be defined for all "exceptional" operands (see 1.8.).

## 3. Arithmetic Comparisons

Conclusion:

Since the arithmetic relations (<, >, =, etc.) are exactly defined for all representable numbers, they must always be correctly computed without causing exception-conditions (it is not acceptable to risk causing overflow or underflow by performing a subtraction). Also the results of such comparisons must be defined for all "exceptional" operands (see 1.8.).

---

There was no time left for further discussion.

The chairman asked those present to submit to him in writing their views on all topics in the list (at the beginning of this summary), whether discussed so far or not.