ISO/IEC JTC1/SC22
Languages
Secretariat: CANADA (SCC)

ISO/IEC JTC1/SC22

N851

OCTOBER 1990

TITLE : Summary of Voting and Comments Received on a proposal to register document N796 as a Committee Draft on: Language Compatible Arithmetic Standard

SOURCE : Secretariat JTC1/SC22

WORK ITEM : JTC1.22.28

STATUS : New

CROSS REFERENCE : N797, N796

DOCUMENT TYPE : Summary of Voting

ACTION : For information to SC22 Member Bodies. See Attached.

Address reply to: ISO/IEC JTC1/SC22 Secretariat
J.L. Côté, 140 O'Connor St., 10th Floor
Ottawa, Ont., Canada K1A 0R5
Telephone: (613)957-2496 Telex: 053-3336
Fax: (613) 996-2690

SUMMARY OF VOTING ON:

Letter Ballot Reference No: SC22 N797
Circulated by :JTC1/SC22
Circulation Date :1990-06-18
Closing Date :1990-09-28

SUBJECT: Proposal to register document N796 as a Draft Proposal on Language Compatible Arithmetic Standard

---

The following responses have been received:

'P' Members supporting the proposal , without comments : 07 SEE ATTACHED.

'P' Members supporting the proposal, with comments : 02 SEE ATTACHED.

'P' Members not supporting the proposal: 00

'P' Members abstaining :00

'P' Members not voting: 11 (see list)

Comments:

Attachment 1 - France
Attachment 2 - Netherlands

Secretariat Action:

The comments will be submitted to WG11 for consideration. Based on WG11's recommendation, after a review of the comments, the Secretariat will proceed with the registration of document N796, or a revised version of the document, as a Committee Draft.

ISO/IEC JTC1/SC22 LETTER BALLOT SUMMARY

PROJECT NO: JTC1.22.28

SUBJECT: Proposal to register N796 as a Draft Proposal (Language Compatible Arithmetic Standard)

Reference Document No: N796 **Ballot Document No:** N797

Circulation Date: 1990-06-18 **Closing Date:** 1990-09-28

Circulated To: SC22 P,O,L **Circulated By:** Secretariat

SUMMARY OF VOTING AND COMMENTS RECEIVED

	Approve	Disapprove	Abstain	Comments	Not Voting
'P' Members					
Austria	( )	( )	( )	( )	(x)
Belgium	( )	( )	( )	( )	(x)
Canada	( )	( )	( )	( )	(x)
China	( )	( )	( )	( )	(x)
Czechoslovakia	( )	( )	( )	( )	(x)
Denmark	(x)	( )	( )	( )	( )
Finland	(x)	( )	( )	( )	( )
France	(x)	( )	( )	(x)	( )
Germany F.R.	(x)	( )	( )	( )	( )
Hungary	( )	( )	( )	( )	(x)
Iran	( )	( )	( )	( )	(x)
Italy	(x)	( )	( )	( )	( )
Japan	(x)	( )	( )	( )	( )
Netherlands	(x)	( )	( )	(x)	( )
New Zealand	( )	( )	( )	( )	(x)
Sweden	( )	( )	( )	( )	(x)
Switzerland	( )	( )	( )	( )	(x)
UK	( )	( )	( )	( )	(x)
USA	(x)	( )	( )	( )	( )
USSR	(x)	( )	( )	( )	( )
'O' Members					
Australia	( )	( )	( )	( )	( )
Brazil	(x)	( )	( )	( )	( )
German Dem Rep.	( )	( )	( )	( )	( )
Iceland	( )	( )	( )	( )	( )
India	( )	( )	( )	( )	( )
Korea	( )	( )	( )	( )	( )
Norway	( )	( )	( )	( )	( )
Poland	( )	( )	( )	( )	( )
Portugal	( )	( )	( )	( )	( )
Singapore	( )	( )	( )	( )	( )
Turkey	( )	( )	( )	( )	( )
Thailand	( )	( )	( )	( )	( )
Yugoslavia	( )	( )	( )	( )	( )

attachment 1
to N851

AFNOR COMMENTS ON ISO/IEC JTC1/SC22/N796

A LANGUAGE COMPATIBLE ARITHMETIC STANDARD

First page 1, Notes on potential changes to the language compatible standard

1 The proposed document does not reflect exactly the decision of the May/June meeting of WG11 as shown in the minutes, section 8.1, paragraph 3 ("a motion to forward N167 changed as noted ..."); the document has not been changed, but a note, whose effect is not clear, has been sent instead. In particular, it may be difficult for some readers to relate references in the Notes with the text of the main document; it is the case for "optional operations".

Page 3, Foreword

The Foreword should be reviewed; in particular:

2 - the discussion about the NWI is irrelevant;

3 - warning should be given that NOTES along the text are informative only.

Page 5, 1.1 Specifications included in this Standard, paragraph 2

4 Denormalized is specified in terms of other non ISO Standards. The full definition should be given. The reference should be deleted. If not, it should be dated.

Page 6, 1.2 Possible extensions to this Standard

5 This paragraph is irrelevant in a Standard and should deleted.

Page 6, 1.3 Specifications not within the scope of this Standard, last paragraph

6 The term "exponent bias", in the last sentence, is not well known, and would deserve a definition.

Page 7, 2 Conformity

Conformance rules should be given to specify how a standard implementation of a language can conform to LCAS; for example:

7 - How can a standard implementation of a language also comply to LCAS, when the Standard of the language does not support all of the arithmetic types defined in LCAS; for example, a conformity statement could be added to specify that a LCAS conforming implementation need not to support those arithmetic types not defined in the language Standard, but must support according to the LCAS Standard those arithmetic types that are defined in both the language and the LCAS Standards;

8 - How can a standard implementation of a language also comply to LCAS, when the Standard of the language does support other arithmetic types (complex, fixed scaled, ... data types) than those defined in LCAS; for example, a conformity statement could be added to specify that the implementation of data types permitted by a language Standard, but not defined in the LCAS Standard, does not render the implementation LCAS-non-standard;

9 - How can a standard implementation of a language also comply to LCAS, when the Standard of the language does not support all of the operations (square root, ...) defined in LCAS;for example, a conformity statement could be added to specify that a LCAS conforming implementation need not to support those operations not defined in the language Standard, but must support according to the LCAS Standard those operations that are defined in both the language and the LCAS Standards;

10 - How can a standard implementation of a language also comply to LCAS, when the Standard of the language does not allow extensions to the language and does not provide all of the intrisic inquiry functions (MAXINT, ...) needed to support LCAS; for example, a conformity statement could be added to specify that the implementation of operations permitted by a language Standard, but not defined in the LCAS Standard, does not render the implementation LCAS-non-standard

See also comments below on section 5.

Page 8, 3.2 Definitions, 8. Rounding function

11 It should be said that *X* is a discrete subset of *R* since in a given finite interval of *X* there is only a finite number of values.

Page 13, 4.2 Floating point types

12 In general, specific hardware should not be excluded by the Standard. In particular, floating point data whose representation is in 2's complement should be permitted, and therefore the definition of the *neg* and *abs* operations should be able to return *overflow* (as it does for integers).

Pages 14, 15, 16, 4.2 Floating point types, chk_F

13 The definition of chk_F makes the definition of the operations on page 16 unnecessarily complex. Defining a new "range checking" function ch_F as

$$ch_F : R \rightarrow F \cup \{overflow, underflow\}$$

by (or equivalent to)

$$ch_F(x) = chk_F(x, rnd_F(x))$$

would simplify subsequent definitions on page 16, e.g. add_F would become:

$$add_F(x, y) = ch_F(add^*_F(x, y))$$

Page 18, 4.2 Floating point types, last paragraph

14 What does "*it is recommended*" mean? Either change the sentence to read "F_1, F_2, F_3, ... *should also satisfy* ...", or make a NOTE of the paragraph.

Page 19, 4.3 Conversion operations

15 Third alinea from the end, "shal" should read "shall".

Pages 19-20, 5 Relationship with language Standards

16 This section pertains to section 2, Conformity (see comment above).

17 The second paragraph is somewhat debatable. It says that if a standard implementation of a language is conflicting with LCAS, a special mode of execution should be provided. But first, under that mode of execution, the implementation is no longer a standard implementation of the language. And second, no definitions exist of the side effects on the overlapping areas resulting from applying the special mode of execution

18 Starting from the third paragraph, much of the text is example. It should be formally recognized as such.

19 In addition, these examples refer to languages as defined in specific Standards. These specific Standards should be indicated, either here, or in the Foreword.

20 Again in these examples, references are given to not yet standardized languages, such as LISP. Such references should be deleted.

21 The last paragraph page 20 provides a list of non supported features. This list and those of sections 1.2 and 1.3 should be gathered together.

Page 23, A

22 Page 23 is missing.

Pages 23, 48, 51, 57, and 59

23 A, B, C, D, and E should be clearly identified as Annex A, Annex B, ... so that the annexes referred to in the last paragraph of the Foreword, page 3, are precisely Annexes A, B, ...

Same comment on page 1 and 2, Contents.

Page 25, A.1 Scope, last sentence

24 It is stated that "each primitive operation contains at most one rounding error"; this is not true in the case of "pre-rounding" (see second NOTE on page 15).

Page 26, A.1.2 Possible extensions to this Standard

25 The second sentence of the last paragraph of the page ends with "this standard". It would be clearer to say "this LCAS standard" so that the reader is not confused between Pascal and LCAS.

26 The last sentence of the same last paragraph refers to *ulp* whereas this operation is only explained later on page 35, item 10.

Page 26, A.1.2 Possible extensions to this Standard
Page 31, A.4.1 Integer types

27 Complete references (number, date, ...) should be given to the Standards Pascal, Ada, Modula-2, ... , referred to in the examples.

Page 27, A.1.3 Specifications not within the scope of this satndard

28 The reference to [14] in the second paragraph does not refer to the right Standard (see page 57).

Page 27, A.2 Conformity, first paragraph

29 Can "conformity consist of an implementation ... together with documentation"? It would probably be better to say that "a conforming system consists of an implementation ... together with documentation ...".

Page 28, A.2 Conformity, Validation

30 "Conformitym" should read "conformity" (first paragraph).

31 The title of ISO-9001 should be reminded (third paragraph).

Page 28, A.3 Notations and definitions

32 The third sentence of the first paragraph says: "if the standard were entirely written in English";it would be more precise to say: "if the standard were entirely written in an unformal way, e.g. in a natural language like English,".

Page 29, A.3.1, Notation

33 Last sentence says that "other mathematical symbols are defined as they are needed in the text". Some definitions are missing, such as:

- x ∈ *R* means x is in the set *R*;
- (a, b) means the set with only a and b as members;
- F_1 U F_2 means the set of members of either or both of F_1 and F_2;
- (x ∈ *R* | ...) means the set of all members x of *R* for which ... is true.

Page 29, A.3.2 Definitions

34 In the discussion of "notification" and "exception", a reference should be given to the Technical Report on the preparation of programming language Standards, where the term is used, and an harmonization should be attempted between the Technical Report and the LCAS Standard.

35 Same comment as above on the third paragraph: *X* is a discrete subset of *R*.

36 Same comment as above about mathematical symbols:

- the symbol => may be read "implies".

Page 31, A.4.1 Integer types

37 At the bottom of the page, in the definition of mod_I, $\leq y$ should read $< y$

38 "NaNs" referred to in the third full paragraph of the page need to be defined.

Page 32, A.4.2 Floating point types
Page 36, A.4.2.2 Rounding and checking

39 Incomplete reference to IEEE (no reference to Annex D).

Page 33, A.4.2 Floating point types

40 In the fifth paragraph of the page, do not split "all" over two lines.

41 References are given to specific vendor implementations. These references should be deleted, as they are discriminatory, and may become out of date at any time.

Page 34, A.4.2 Floating point types

42 It should be reminded that the geometric mean referred to in item 5 is the square root of the product.

43 The foot note refers to the table on the previous page.

Page 35, A.4.2.1 Floating point operations

44 Item 3, "in-range" should read "in range".

45 Item 12, could not "ties" be replaced by a more common word (at least for non English readers)?

Page 40, A.4.2.5 Levels of predictability

46 Item 3, though the term "model" can be understood, it does not seem the appropriate word in the context.

47 Item 4, change "were" to "are" and "satisfied" to "satisfy".

Page 42, A.4.2.6 Identities

48 One more notation is to be explained:

- [1, r) means a range that includes 1 but excludes r.

Page 47, A.7 Documentation requirements

49 Last paragraph, last line, "aways" should be "ways".

Pages 25-47, A

50 Accepted comments on the body of the Standard should be reflected in this Annex A.

Pages 48-50, B Suggested parameter names

51 The language Standards referred to in this annex should be precisely identified (number, date, ...), and references to ISO Standards should be given rather than references to (even equivalent) other Standards.

Page 51, C Example conformity statement

52 Fortran 8x: reference should be replaced by a standard DIS, IS, ... number.

53 IEEE 754 Standard should specify a refence to Annex D.

Page 58, E Glossary

54 **Axiom:** "rules" should read "rule".

55 **Denormalized:** "result from" may be better understood than "be due to".

56 **Exception:** this term has not been used; section 6 uses "violation" instead; see also other comment above on A.3.2.

Kalfjeslaan 2
Postbus 5059, 2600 GB Delft
Telefoon (015) 690 390
Telefax (015) 690 190
Telex 38 144 nnl nl

Attachment 2
to N 851

Comment on the Working Draft of
A Language Compatible Arithmetic Standard
Document ISO/IEC JTC1/SC22/N796

We are in favour of registering this document as as draft proposed standard. We feel that the development of future languages and language standards will benefit from this work. We hope that the comments below will help to improve the document.

a) The definition of the operations seem to be used with to goals in mind:

1) to describe the static behaviour as classic mathematical functions;

2) to describe the dynamic behaviour.

To achieve this effect a mapping is used in which the operations produce a single value, either numeric or exceptional. This attempt to coerce both static and dynamic behaviour into a single codomain has a few undesired effects.

Example 1
Paragraph 6.2, item 2 says that an implementation is allowed to continue after notification with a *well defined in-range value* in place of the exception. Suppose that an implementation continues with the value zero after an overflow of *add*$_I$. What then is the result of the *add*$_I$ function? Zero or overflow?

Example 2
The standard does not define the term *in-range* in §6.2. The codomain of the *add*$_I$ function is $I \cup \{\text{overflow}\}$. Is overflow in-range or not?

Example 3
Can an implementation get away with using the same value for overflow, underflow, zero_divide and undefined?

It would perhaps be more appropriate to define the operations as functions that produce tuples with two elements: the resulting value and the dynamic behaviour.

$$add_I : I \times I \rightarrow I \times \{ok, overflow\}$$

b) The document does not allow for the standardised use of 'modulo' arithmetic.

c) The description of the types in ANSI-C is not correct(§B.2). The unsigned integer types can not conform to the standard. The operations on values of this type use 'modulo' arithmetic and can not overflow. The only exception possible with this type is zero-divide. (ANSI-C standard, X3.159-1989, page 24)