ISO / IEC JTC1 / SC22
Programming languages, their environments and system software interfaces
Secretariat: CANADA (SCC)

ISO/IEC JTC1/SC22
N 1325

FEBRUARY 1993

TITLE:	Summary of Voting and Comments received on CD 10967-1: Information Technology - Programming languages, their environments and system software interfaces - Language compatible arithmetic
SOURCE:	Secretariat ISO/IEC JTC1/SC22
WORK ITEM:	JTC1.22.28
STATUS:	New
CROSS REFERENCE:	N1263
DOCUMENT TYPE:	Summary of voting - CD
ACTION:	For information to SC22 Member Bodies. The comments have been submitted to WG11 for consideration and recommendation on further processing of CD 10967-1.

Address reply to: ISO/IEC JTC1/SC22 Secretariat
J.L. Côté
Treasury Board Secretariat
140 O'Connor St., 10th Floor, Ottawa, Ontario, Canada, K1A 0R5
Tel.: (613)957-2496 Telex: 053-3336 Fax: (613)957-8700

SUMMARY OF VOTING ON:

Letter Ballot Reference No:	SC22 N1263
Circulated by	:JTC1/SC22
Circulation Date	:1992-10-01
Closing Date	:1993-01-28

SUBJECT: CD 10967-1: Information Technology - Programming languages, their environments and system software interfaces - Language Independent Arithmetic

The following responses have been received:

'P' Members supporting CD,	without comments	: 0915
'P' Members supporting CD,	with comments	: 01
'P' Members not supporting the CD		: 01
'P' Members abstaining		: 00
'P' Members not voting		: 11

Secretariat Action:

The Secretariat will forward the attached comments to WG11 for consideration and recommendation from WG11 on further processing of CD10967-1.

ISO/IEC JTC1/SC22 LETTER BALLOT SUMMARY

PROJECT NO:	JTC1.22.28
SUBJECT:	SECOND CD 10967-1: Information Technology - Programming languages, their environments and system software interfaces - Language Independent Arithmetic

Reference Document No: N1263	Ballot Document No: N1263
Circulation Date: 1992-10-01	Closing Date: 1993-01-28
Circulated To: SC22 P, O, L	**Circulated By:** Secretariat

SUMMARY OF VOTING AND COMMENTS RECEIVED

	Approve	Disapprove	Abstain	Comments	Not Voting
'P' Members					
Austria	( )	( )	( )	( )	(✓) (NOT COUNTED)
Belgium	(x)	( )	( )	( )	( )
Brazil	( )	( )	( )	( )	(✓)
Bulgaria	( )	( )	( )	( )	(✓)
Canada	(x)	( )	( )	( )	( )
China	(x)	( )	( )	( )	( )
Czechoslovakia	( )	( )	( )	( )	(✓)
Denmark	(x)	( )	( )	( )	( )
Finland	( )	( )	( )	( )	(✓)
France	(x)	( )	( )	( )	( )
Germany	(x)	( )	( )	( )	( )
Greece	( )	( )	( )	( )	(✓)
Iran	( )	( )	( )	( )	(✓)
Italy	(x)	( )	( )	( )	( )
Japan	( )	( )	( )	( )	(✓)
Netherlands	(x)	( )	( )	( )	( )
New Zealand	( )	( )	( )	( )	(✓)
Romania	( )	( )	( )	( )	(✓)
Sweden	(x)	( )	( )	( )	( )
Switzerland	( )	( )	( )	( )	(✓)
UK	(x)	( )	( )	(x)	( )
USA	( )	(x)	( )	(x)	( )
USSR	( )	( )	( )	( )	(✓)
'O' Members					
Argentina	( )	( )	( )	( )	( )
Australia	( )	( )	( )	( )	( )
Cuba	( )	( )	( )	( )	( )
Hungary	( )	( )	( )	( )	( )
Iceland	( )	( )	( )	( )	( )
India	( )	( )	( )	( )	( )
Korea	( )	( )	( )	( )	( )
Poland	(x)	( )	( )	( )	( )
Portugal	( )	( )	( )	( )	( )
Singapore	( )	( )	( )	( )	( )
Turkey	( )	( )	( )	( )	( )
Thailand	( )	( )	( )	( )	( )
Yugoslavia	( )	( )	( )	( )	( )

D IS C

UK COMMENTS ON CD 10967-1:1992 LANGUAGE INDEPENDENT ARITHMETIC -
PART 1:INTEGER AND FLOATING POINT ARITHMETIC

The UK, noting that there are no known significant technical objections to the content of this document, and believing that it is valuable to users of programs and important to the development of Programming Language standards votes YES to DIS registration of this document, with the following comments.

Comments

The following changes to the introduction, under Goals, should be made:

1. In the first goal replace the two occurrences of "portability" with "predictability".

   Rationale: Portability, in a bit-wise sense, is not achievable. The proposed change better describes this aim of the standard, and is achievable.

2. Replace the second sentence of the first goal with:
   "The focus is on assuring users of single programs in specific programming language of a documented level of accuracy of arithmetic operations on which they can rely, and the portability of such programs across platforms."

   Rationale: This change gives needed further emphasis to the value of the standard to users of programs, whether or not they are ported across platforms.

3. In the second paragraph replace "Our second goal" with "The second goal of this international standard".

For the benefit of many numerical users, LIA-1 should be accompanied, in an informative annex, by a specific binding to IEC 559 (IEEE 754). In addition, definitions should be adapted so that they are consistent with IEC 559 (IEEE 754).

JMS/MO
18 December 1992

U.S. Position on the Second CD Ballot for ISO/IEC CD 10967-1

Language-independent arithmetic --
Part 1 Integer and floating point arithmetic

The United States votes NO on the second LIA-1 CD ballot. Our vote would be changed to YES upon adoption of major recommendations 2.1, 3.1, 4.1, 4.2, and 4.3 below. The U.S. will offer an additional contribution with a detailed rationale and point-by-point description of the recommended changes, including suggested text for each.

The United States position is that LIA-1 need not be abandoned due to a perceived conflict with the IEEE 754 (IEC 559) standard. However, it is necessary to modify the document significantly in order to achieve consensus. To do this, the major areas that need to be addressed are:

A clear statement of purpose that positions LIA-1 as a bridge between existing arithmetic implementations and programming language standards and does not raise expectations of ensuring source code portability that such a standard cannot fulfill.

Making compliance effective at the level of datatype rather than system to avoid giving the appearance of a language standard.

Relaxation of conformance requirements that have given LIA-1 the appearance of a hardware standard and have made some architectures (Cray in particular) non-conforming. This relaxation must be accompanied by increased documentation requirements that will clearly expose the functional shortcomings of such architectures.

Extending integer conformance to modular arithmetic datatypes.

Reduction of the mathematical complexity of the document so as to make it more accessible to the large numbers of groups that it affects. (This will follow in large part from the simplifications cited above, although there remains considerable expository work as well.)

Addition of capabilities and exposition that clearly demonstrate added value to IEEE 754 implementations.

R E C O M M E N D A T I O N S

1. GOALS AND APPROACH

1.1 Restate the goal of the standard in terms that avoid use of the word "portable". Distinguish "portability" as something that can be preplanned as code is written or coincidental afterwards. Make clear that LIA-1 aims to enhance the former endeavor but will not significantly aid the latter. Make clear that the constraints imposed by LIA-1 are aimed at making the preplanned task easier (by requiring written documentation and provision of parameters and useful functions that can be economically implemented in software), but the constraints on hardware concern notification and value sets of datatypes only, and not operations on the datatypes.

1.2 Acknowledge that in order to include Cray, the range of "bizarre" possible architectures is thereby increased enormously. The penalty such architectures must pay is increased documentation requirements and possible adverse public relations that arise from this documentation. (The aim is to illuminate rather than eliminate.)

1.3 Make clear that LIA-1 intends to facilitate identification and illumination of deviations from IEEE 754. For other floating point datatypes, the LIA model provides a shorthand that makes documentation for model-conforming arithmetics easier to provide. For datatypes that do not fit the model (Cray particularly), conformance is still attainable provided the deviations are properly documented and the appropriate set of functions is supported.

1.4 Emphasize there is no intent to define an "LIA machine". It therefore makes no sense to require programs to certify consistent or "correct" behavior on all conforming implementations.

1.5 Make clear that the role of LIA-1 is to provide a bridge between a specific arithmetic implementation (such as IEEE 754) and a language standard. In order to be useful, it operates as a bridge between IEEE 754 AS WELL AS OTHER IMPLEMENTATIONS to bindings to specific languages.

1.6 Include an example of how to use the LIA-1 dynamically available parameters in a program that exploits them to choose an appropriate algorithm based on the available capabilities. This can be done in an annex in conjunction with an example of a specific binding.

2. CONFORMANCE -- GENERAL

2.1 MAJOR - Change the conformance concept to apply to datatypes, not to systems. In particular, decouple integer and floating point conformance. In order to do this, it must be made clear that the integer types implied in Section 4.2 need not be conforming types.

2.2 Tighten emax and emin parameter constraints to make all floating point parameters representable in their own type.

3. INTEGER

3.1 MAJOR - Extend the recognized variants of the integer type to add "modular" to "bounded" and "unbounded". The latter two behave as at present. The modular variant is like bounded but does all arithmetic modulo (maxint - minint + 1). It does not overflow, but does remain subject to "undefined" for zero divisors.

4. FLOATING POINT

4.1 MAJOR - Include less regular architectures (such as Cray) in the following way. For each of the six major operations (add, subtract, multiply, divide, scale, and convert), require three parameters:

round_error (a floating point number that represents the maximum number of units in the last place that a normalized result can be in error), input_perturb (a floating point number that represents the maximum number of units in the last place that operands would have to be perturbed separately in order to obtain a precise result), and

model_flag (a Boolean that specifies whether the properties of the version 4.0 LIA floating point model hold).

If model_flag is true, then a program can rely on maximum rounding_error being less than 1 ulp, all existing axioms being true, and can further interrogate the parameter round_style whose values come from the set {round to nearest ties to even, round to nearest ties away from zero, round towards zero, other}. If model_flag is false, then the axioms need not be true, but ALL DEVIATIONS MUST BE DOCUMENTED.

In making these changes, WG11 should also consider the possibility that the overflow and/or underflow threshold for each operation may not be equal to the representation thresholds defined by the model and may additionally be different from one operation to the next.

4.2 MAJOR - Add a Boolean flag to indicate whether the datatype is IEEE 754 conforming (formats AND arithmetic).

4.3 MAJOR - Correct the typographical error in the discussion of underf low that implies that IEEE 754 is still not supported due to divergent requirements on underflow detection.

5. CLARIFICATIONS

5.1 Codify the various uses of Boolean types in the standard to make clear that the implementation must support some encoding of "true" and "false" in some datatype.

5.2 Provide text that explicitly enumerates all hardware constraints.

5.3 Define div and rem twice in Section 4.1 in order to avoid introducing the function rndI.

5.4 Restructure Section 4.2, reduce overloaded nomenclature, try to remove the functions (rndF, add*, and ResultF) that contribute to the exposition but are not required of implementations, and clarify the roles of any such "helper" functions that remain.

5.5 Reword the "range of translations" discussion to show how unary and binary operations may be composed in larger expressions -- use A*B + C*D as the example.

5.6 Try to remove terms such as "prompt", "hard-to-ignore", "error" and "exceptional" from the discussion of notification.

5.7 Make changes to the informative annexes to reflect changes to the normative text. Add clarifications to answer specific U.S. public review comments that indicated confusion. Annex B need not be extend to provide suggested bindings to IEEE 754.

5.8 Add text to enumerate and summarize all normative requirements.

6. PROCESS

6.1 In view of the long lead times entailed, SC22 should start the process now of creating bindings between specific languages and the LIA-1 and IEEE 754 standards.