ISO/IEC JTC1/SC22/WG11 N 81

Proposal for an International Standard for

Floating Point within a Programming Language

B A Wichmann, 87-11-19

Introduction

Does 2.0 + 2.0 = 4.0? The answer is 'perhaps'. The reason for the uncertainty is that there is no standard for floating point, as widely used on computers, which would guarantee this. Of course, since floating point is by its nature approximate, the question is a little unfair, but other less obvious questions can be asked with no guarantee of a 'sensible' result. All this is a little worrying when one takes into account the reliance placed upon the results computed using floating point via a high-level language. This note proposes that this uncertainty should be removed by means of a suitable new floating point standard. Uncertainties will still arise when poor algorithms are used, but these are addressed by the use of numerical analysis.

Objectives

The proposed standard should meet the following objectives, in decreasing order of priority:

1. The requirements of the standard shall imply the properties of floating point needed by numerical analysts to deduce the correctness of conventional algorithms.

2. The standard shall be formulated in such a way as to permit the majority of existing and future computer systems to claim conformance to the standard.

3. The standard shall not conflict with the requirements of the main programming language standards (FORTRAN, Ada, Pascal, etc).

4. The standard shall be formulated in such a way as to allow conformance by future high-performance systems which use pipelining and concurrency.

5. The standard shall allow rigorous conformance testing of processors.

It is admitted that there are some conflicts between these objectives. Some suppliers have sacrificed accuracy for speed. However, the work of W S Brown has shown how some of these objectives can be reconciled.

Machines are different in the basic characteristics of their floating point and therefore such a standard would have to be parametrised (for instance, by giving the radix, mantissa length and exponent range).

Relationship with other Standards

The only current international standard for floating point is an IEC one. This is based upon a draft of the IEEE standard (P754), and hence does not represent a complete standard.

The IEEE binary floating point standard does not meet objectives 2, 3 and 4 above. It is a standard for a particular form of floating point. Conformance to this IEEE standard should imply conformance to the standard proposed here (to satisfy objective 2).

The IEEE generalised radix floating point standard does not meet requirements 2 and 3 either, since (for instance) a radix of 16 is not permitted, nor is truncation rather than rounding.

Conflicts arise between the IEEE standards and programming languages since languages do not (except by an extension) allow numeric overflow and underflow to be trapped in such a way that the expression being computed can be re-computed. Similar problems arise with high performance machines for which the processing of exceptional events can cause the pipeline to be flushed or the processor to wait.

In short, this proposal can be seen as being complementary to the existing IEEE standards and to the programming language standards. The IEEE standards and the hardware specifications used by suppliers are essentially product specifications while this proposal is a requirement specification which can be satisfied in a number of logically distinct ways. Since floating point is accessed via a high-level language, this proposal can be viewed as a supplement to a language standard.