A Response to the comments from IFIP Working Group 2.5
on ISO/IEC 10967:1991 (version 3.1)
Brian Wichmann 92-02-06

I agree with the comments which encourages the use of the IEEE 754 floating point standard. There is no intention of undermining this standard with ISO 10967. Indeed, as Kahan has noted, the documentation requirements of 10967 is a useful compliment to an IEEE system.

The third paragraph appears to suggest that the exponent range requirements in the standard should be changed. However, no proposal is made and therefore it is hard to know how to react. Clearly, whatever the exponent range is, it may be too narrow for some applications. I would welcome a proposal here. However, ISO standards are based upon a consensus. It seems doubtful to me that a consensus could be obtained which would rule out IBM (HEX) double precision due to a restricted range, especially since some numerical analysts would claim that IBM single length has inadequate precision.

The paragraph which refers to Kahan seems to be based on a misconception of the goals and objectives of 10967. As a thought experiment, consider a world in which every machine had IEEE 754 floating point. Hence I claim 10967 would perform useful role for two reasons:

a) it covers integer arithmetic and numeric conversions;
b) it covers the semantic gap between programming languages and IEEE 754.

To illustrate this later point consider the example given in the IFIP comments written in Pascal (say):

```
x,y,z: real;
...
z := x / sqrt(sqr(x) + sqr(y));
```

Contrary to the statement in the comment, this can yield a value greater than 1.0 for z. If x,y and z are single precision, and all the computation is performed in single precision, then underflow can occur, which either assigns plus or minus infinity to z, or is detected as an error. Tests undertaken at NPL shows that the functions sqr and sqrt could be computed to extended or double precision, giving quite different results. All one can say is that it is more likely that the operator + is computed with precision greater than single. These variations actually arise on a Sun/3 system with different compilers and compiling options.

Kahan in his comments notes that the documentation requirements of 10967 would be useful. Moreover, I would agrue that the ‘code-bloat’ objections to 10967 also applies to IEEE unless one knows exactly how the IEEE hardware is exploited by the compiler. My experience is that actual compiler documentation is inadequate here.

The paragraph which objects to the handling of notification has been noted by WG11. The current proposals for the revision of this part of 10967 will, in my view, answer these criticisms. It would be very helpful if IFIP WG2.5 could review the next draft and see if they are satisfied. (I don’t go into the details of the proposed changes here for fear that a mis-represent it and cause confusion -- the next draft is the right place to look!)

The last paragraph would appear to suggest that Cray arithmetic should conform but this is not clear. I disagree strongly with the last sentence. There is no desire to more away from the strictest standard, indeed I would agrue that an IEEE system documented in the way 10967 requires IS the strictest reqirement that we can currently seek.