

**Document:** N1515  
**Date:** 2010/09/23  
**Author:** Jim Thomas  
**Subject:** Fix for Annex F pow special case

In F.10.4.4 "The pow function", the second bullet implies  $\text{pow}(\pm 0, -\infty)$  must raise the divide-by-zero floating-point exception. This is inconsistent with IEC 60559 which says that divide-by-zero occurs "only when INFINITY is created from finite operands". For example,  $\text{INFINITY} / 0$  does not raise the divide-by-zero floating-point exception. 754-2008 states this as "The divideByZero exception shall be signaled if and only if an exact infinite result is defined for an operation on finite operands." The creation of an infinity from finite operands (via overflow of divide-by-zero) raise floating-point exceptions, but valid operations on infinities do not.

The following change fixes the defect that pow is now specified to raise a floating-point exception when the indicated condition didn't actually occur. Also, it supports consistency with IEC 60559, which is the point of Annex F.

**Proposed changes:**

In the second bullet in F.10.4.4 change

- $\text{pow}(\pm 0, y)$  returns  $+\infty$  and raises the "divide-by-zero" floating-point exception for  $y < 0$  and not an odd integer.

to

- $\text{pow}(\pm 0, y)$  returns  $+\infty$  and raises the "divide-by-zero" floating-point exception for finite  $y < 0$  and not an odd integer.

Insert another bullet after the second bullet:

- $\text{pow}(\pm 0, -\infty)$  returns  $+\infty$ .