Proposal for C2Y
WG14 N3703

Title:	Preferred quantum exponent
Author, affiliation:	C FP group
Date:	2025-07-15
Proposal category:	Editorial
Reference:	N3550

Background

This proposal addresses three issues regarding preferred quantum exponents, the first two raised by Vincent Lefèvre in email [cfp-interest 3506] and a related issue.

Issue 1: 5.3.5.3.4 #9 says

> … Table 5.2 gives rules for determining preferred quantum exponents for results of ISO/IEC 60559 operations, and for other operations specified in this document. When exact, these operations produce a result with their preferred quantum exponent, or as close to it as possible within the limitations of the type. When inexact, these operations produce a result with the least possible quantum exponent. …

This specification is fine for ISO/IEC 60559 operations and others that are required to be correctly rounded, because it is reasonable to require implementation with correct rounding to determine whether the result is inexact. But Annex F does not require correct rounding for all `<math.h>` functions, and implementation that does not round correctly might not be able to determine whether a result is exact. This is not a problem for the trigonometric, hyperbolic, and exponential and logarithmic functions because they are transcendental and cannot possibly be exact except for the few inputs for which Annex F specifies exact results. However, the power functions (other than `sqrt` which is required to be correctly rounded) might inadvertently achieve an exact result. The specification above does not cover such cases. What is the preferred quantum exponent when exactness is undetermined?

Issue 2: 5.3.5.3.4 #10 says

> … how the preferred quantum exponents of the operands, $Q(x)$, $Q(y)$, etc., determine the preferred quantum exponent of the operation result, …

However, $Q(x)$, $Q(y)$, etc. denote actual quantum exponents, and these are what determine the preferred quantum exponents of operation results. The first "preferred" in the quote doesn't make sense and needs to be removed.

Issue 3: In 5.3.5.3.4 Table 5.2 the two rows for conversion from non-decimal floating type reflect a misunderstand of preferred quantum exponents. The preferred quantum exponent applies only when the result is (known to be) exact. When the result is inexact the quantum exponent of the result is always the least possible, a fact that does not belong in the table. Only one row is needed, as with the other operations in the table.

Suggested Changes

In 5.3.5.3.4 #9, change:

> … When inexact, these operations produce a result with the least possible quantum exponent. …

to:

> … When inexact, and when the implementation does not determine exactness*), these operations produce a result with the least possible quantum exponent. …
>
> *) It is assumed the implementation will determine exactness if the operation is specified to be correctly rounded or if a particular numeric result is specified for the argument. In other cases, the implementation can fail to determine exactness. For example, an implementation can inadvertently compute the exact result value for `rootnd32(.125DF,3)`, yet return (5000000, −7) as though the result were inexact, instead of (5, −1) which would be the result if exactness were determined. The special cases for the transcendental functions in F.10.2, F.10.3 and F.10.4 cover all the cases where those functions can be exact.

In 5.3.5.3.4 #10 change:

> … how the ~~preferred~~ quantum exponents of the operands, $Q(x)$, $Q(y)$, etc., determine the preferred quantum exponent of the operation result, …

to:

> … how the quantum exponents of the operands, $Q(x)$, $Q(y)$, etc., determine the preferred quantum exponent of the operation result, …

In 5.3.5.3.4 Table 5.2 change:

~~exact~~ conversion from non-decimal floating type	0
~~inexact conversion from non-decimal floating type~~	~~Least possible~~

to:

conversion from non-decimal floating type	0