WG14 N2636
Meeting notes

C Floating Point Study Group Teleconference

2020-11-25
8 AM PST / 11 AM EST / 4 PM UTC

Attendees: Rajan, Jim, Fred, Mike, Damian, Ian, David H,

New agenda items:
WG14 reflector message 18590 response - Added to "Other Issues"

Carry over action items:
None.

Last meeting action items (all done unless specified otherwise below):
Fred: Look into other places to update references to DEC_EVAL_METHOD as per N2546.
Rajan: Discuss with JeanHyde on what to do for N2558 (and mention our plans for N2559).
Jim: Update N2559 to add 'superseded' and update the bibliography as per Josephs comments.
Jim: Get an N# and submit the new draft of TS Part 3 as an annex.
Fred: Rework CFP 1797 to address the comments in the message.
Fred: Write a paper to remove the fminfN/fmaxfN/fmindN/... functions from part 3 as an annex.
Fred: Create a paper for the missing cases for compound{n} as per CFP 1793.
Jim: Submit the need for editorial changes along the lines of CFP 1794 to JeanHyde.

New action items:
Fred: Redo updates to N2546 with the changes in CFP 1859.
Jim: Give a short summary of differences between N2579 and N2601 (differences between TS part 3 as an Annex updates 2 and 3).
Fred: Update the example in G.5.1 using fmax to use the newer functions as a new proposal.
Rajan: Respond to the WG14 reflector message to say CFP wants equivalence to strtod and hence we don't want to parse digit separators either.
Fred: Write a WG14 editorial informational paper as per CFP 1821.
Fred: Write a CFP paper for the pow(1,NaN) and the compound(NaN, 0) case with respect to quantum exponent of the result.
David H: Look into 'numerically equivalent' vs 'numerically equal' usage in the C standard and revisit CFP 1849.
Fred/Jim: Have a statement in the main body of the standard saying opposite signed zeros compare equal.
Fred: Change 'negative' to say 'less than zero' in certain cases in C.
Jim: Send out something to say negative zero and NaN's with a negative sign bit are not negative values in C.
Jim: Reword the signbit description in C.

Next Meeting(s):
Wednesday, January 13th, 2021, 4PM UTC
ISO Zoom teleconference
Please notify the group if this time slot does not work.

C++ liaison:

None.

C23 integration

Latest C2X draft: http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2573.pdfalso as link on CFP wiki

Previous C2X draft: http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2478.pdfPart 1
Part 2
Part 3
Part 4ab
Part 5abcd
IEC 60559:2020 support

Action item details

Fred: Look into other places to update references to DEC_EVAL_METHOD as per N2546. See CFP 1859.

Fred: Agree with the changes in CFP 1859. Will rewrite the paper.

Rajan: Discuss with JeanHyde on what to do for N2558 (and mention our plans for N2559).

Rajan: JeanHyde is good with the functions list in the Annex rework to a parameterized version, and is willing to do the rest of the standard the same way too. Longer term for that, but he is good with it. Expectation is a paper to get WG14 approval from CFP and him, once the standard LaTeX build system is updated to how he wants it.

For the part 3 as an annex, JeanHyde is good with our changes and does not need us to present it to WG14.

We still need to deal with the changes requested (Ex. All the parameterization that doesn't apply to some of the functions listed).

Jim: Update N2559 to add 'superseded' and update the bibliography as per Josephs comments. https://wiki.edg.com/pub/CFP/WebHome/n2600.pdf- C23 proposal - Revised N2559 update for IEC 60559 2020

Looks good.

Jim: Get an N# and submit the new draft of TS Part 3 as an annex. https://wiki.edg.com/pub/CFP/WebHome/n2601.pdf- C2X proposal - TS 18661-3 annex update 3

AI: Jim: Give a short summary of differences between N2579 and N2601 if asked.

Fred: Rework CFP 1797 to address the comments in the message. See CFP 1843.

Fred: Write a paper to remove the fminfN/fmaxfN/fmindN/... functions from part 3 as an annex. See CFP 1837 - 1848.

Fred: Will rewrite the paper.

Jim: Should we leave the functions out of the Annex?

Agreed.

Fred: Should we remove the DFP versions?

Jim: Those went into part 2. It was implemented by HP. I believe Intel does as well.

Ian: I think IBM implemented them.

No consensus to remove them.

Fred: There was a use of fmax in G.5.1 in an example.

Jim: It should be replaced with one of the new minimum/maximum functions.

AI: Fred: Update the example in G.5.1 using fmax to use the newer functions.

Fred: Create a paper for the missing cases for compound{n} as per CFP 1793. See CFP 1845.

Fred: Found other cases as well. Jim said they are covered by a blanket NaN statement.

Jim: For powr needs to be removed. Leftover from editing.

The copysign line needs to be removed/omitted.

Fred: pow(1,nan) compound(nan,0) both give 1. Everywhere else NaN's give NaN.

754 never says what the preferred exponent of a NaN is, so what is the preferred exponent for that 1?

Jim: Was there an Inf that should have been an infinity symbol?

Fred: No, it was because I don't have an infinity symbol so that was just text to let the editor know.

Jim: Submit the need for editorial changes along the lines of CFP 1794 to JeanHyde.
https://wiki.edg.com/pub/CFP/WebHome/n2602.pdf- C23 proposal - edits for infinity and NaN macros

Jim: The prefix names (D{32,64,128}_) should have been DEC{32,64,128}_ which needs to change. This is handled in N2617.

Other issues

WG14 reflector message 18590 response

AI: Rajan: Respond to the WG14 reflector message to say CFP wants equivalence to strtod and hence we don't want to parse digit separators either.

Fix for decimal SNAN prefixes

Agreed.

Preferred quantum exponents

[Cfp-interest 1829] Re: Preferred quantum exponent Jim Thomas

Jim: Looks to be editorial.

AI: Fred: Write a WG14 editorial informational paper as per CFP 1821.

The quantum exponent for NaN:

David H: It should give the same quantum exponent as the 1 argument in the pow case.

Jim: We'd be getting ahead of 754 here if we said anything.

David: If the other argument doesn't affect the quantum exponent, it should't affect the result either.

AI: Fred: Write a CFP paper for the pow(1, NaN) and the compound(NaN, 0) case with respect to quantum exponent of the result.

Mike: I will do something for 754 to list this as an errata.

Fred: Also issues with extended formats and sign bits of NaN.

Signaling NaNs. See CFP 1833.

Jim: Should we explicitly talk about quiet NaN instead of using the blanket statement that NaNs are implicitly quiet unless stated otherwise.

I think we trip up on this a number of times so I like it.

Rajan: I think if we do a risk/benefit of doing this it seems to balance on the risk side to me (if we miss a case for example).

Fred: There are a number of references F.2.1 that refers to SNAN in math.h that needs to be changed. I will send a email on this.

For the hypot case:

Fred: I don't understand what "numerically equivalent" means.

David H: Why not say numerically equal?

AI: David H: Look into numerically equivalent vs numerically equal usage in the C standard and revisit CFP 1849.

Even zero. See CFP 1826.

Fred: Agree.

Negative zero

[Cfp-interest 1839] Re: Negative zero Damian McGuckin o [Cfp-interest 1840] Re: Negative zero Damian McGuckin

Ian: SNaN's can have an interpretation in the sign bit.

David H: It is allowed for both NaN and SNaN, and up to the implementation.

Ian: Exception handling could be user written code.

David H: It may be user written, but not portable.

Fred: The old IEEE spec says copysign copies the sign bit, but the current IEEE says it is unspecified for NaN's. I need to go back and see where it changed.

Ian: I want C to match 754. This means the sign bit of the NaN be uninterpreted. This is different from saying a NaN does not have any sign.

Jim: 754 shows different levels of abstractions, and in those it shows there is only one QNaN and one SNaN, but at the encoding level it does show the sign. At the representation level in 754 it doesn't have a sign.

Jim: I don't see opposite signed zeros compare equal anywhere in the main body of the standard. If it is not said explicitly, we should say so in Annex F.

Rajan: A simple bitmask AND should be a way to implement the signbit macro.

Jim: That is the intent of IEEE.

AI: Fred/Jim: Have a statement in the main body of the standard saying opposite signed zeros compare equal.

AI: Fred: Still want to change negative to say less than zero in certain cases in C.

David H: Zero's are not positive or negative, but are positive signed or negative signed. NaN's are not positive or negative but are sign bit'd in interchange formats but may not be for extended formats.

I will send it out via email.

AI: Jim: I will send out something to say negative zero and NaN's with a negative sign bit are not negative values in C.

AI: Jim: Need to reword the signbit description in C.

Parameterization of interfaces

Nothing new.

TS 18661 updates. See CFP 1856.

Fred: Does anyone have a copy of IEC 60559:2020? In the past IEC posted something that was different and not correct.