WG14 N2702
Meeting notes

C Floating Point Study Group Teleconference

2021-02-17
8 AM PST / 11 AM EST / 4 PM UTC

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

New agenda items:
See updated agenda. Nothing added during the meeting.

Carry over action items:
David H: Look into numerically equivalent vs numerically equal usage in the C standard and revisit CFP 1849. See [Cfp-interest 1920] "numerically equal" and "numerically equivalent" David Hough

Last meeting action items (all done unless specified otherwise below):
Jim/Fred: Go through all the CFP proposals submitted to ensure they were put in the C standard draft (N2596) correctly. - Not Done.
Fred: Follow up with the WG14 editor about the changes in CFP 1874. - No response.

Fred: Submit CFP 1869 to WG14 with the typo fix of adding a space to the second last change.
Fred: Resend the document relating to updating the example in G.5.1 using max to use the newer functions as a new proposal.
Fred: Submit CFP 1870 to WG14.
Fred: Send out the changed text (CFP 1891) to 754.
Rajan: Test the change in CFP 1891 with IBM's implementation.
Fred: Submit the change proposed in CFP 1891 to WG14.
Jim/Fred: Make CFP 1866 into a draft WG14 proposal with the change "Positive zeros compare equal to negative zeros."
Fred: Change "negative signed value" to "a value with a negative sign" in CFP 1886 and submit it to WG14.
Jim: Make CFP 1881 into a WG14 proposal removing "with a negative sign bit".
Jim: Look at other uses of the phrase "sign bit" in the C standard.
Jim: Submit CFP 1879 to WG14 changing "reports" to "determines" in the footnote.
Rajan: Look into whether "should" can be used in an ISO standard. (Yes, allowed: See https://www.iso.org/foreword-supplementary-information.html)
Jim: Submit CFP 1883 as a WG14 proposal.
Jim: Get permission from WG14 to revise TS 18661-5 along the lines of what is proposed for TS 18661-4 in CFP 1856.
Fred: Follow up with the WG14 editor about the changes in CFP 1874.
Jim: Combine CFP 1880 and CFP 1885, and update CFP 1885 to address the and/or ambiguity in the first suggested change and put it in the form of a proposal.

Jim: Get permission from WG14 to revise TS 18661-5 along the lines of what is proposed for TS 18661-4 in CFP 1856.

Carry over action items for next meetint:

Jim/Fred: Go through all the CFP proposals submitted to ensure they were put in the C standard draft (N2596) correctly.

New action items:

David H.: Look at each use of "numerically equal" and "equivalent" and see what should be changed in the C standard.

David H: Rewrite the conclusion in CFP 1920 so it matches surrounding text and says hypot(x, +-0) is correctly rounded when x is a number along with the CFP 1920 proposed text.

Fred: Make CFP 1896 into a proposal for WG14 removing the word "NOTE" in change 1, and doing the change questioned in 7.31.8.

Fred: Send out the email numbers for handling the CFP 1891 action item

Fred: When presenting CFP 1891 to WG14, ensure that the hypot case is said to not apply.

Jim: Submit the document in CFP 1901 to WG14.

Jim: Submit the paper in CFP 1903 to WG14.

David H: Look to see if setpayload{sig} in IEEE 754 says anything about the sign bit.

Jim: CFP 1908: Change "fk == 0" to "fk = 0".

Jim: CFP 1908: Make a change to put NaNs before infinities: "not floating point numbers, such as NaNs and (signed or unsigned) infinities."

Jim: CFP 1908: Remove the bit-representation paragraph's second sentence.

Rajan: Review Jim's update to CFP 1908 before submission to WG14.

Mike: Bring forward the IEEE errata (CFP 1914) after removing the third item to CFP before bringing it to IEEE.

Fred: See if "cr_" as a prefix is reserved or in the process of being reserved.

Next Meeting(s):

Wednesday, March 24th, 2021, 8 AM PDT / 11 AM EDT / 3 PM UTC
ISO Zoom teleconference
Please notify the group if this time slot does not work.

C++ liaison:

NaN's and Domain errors

C23 integration:

Latest C2X draft: http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2{596,573,478}.pdf also as link on CFP wiki

Part 2
Part 3
Part 4ab
Part 5abcd
IEC 60559:2020 support

Carry over action items:

David H: Look into numerically equivalent vs numerically equal usage in the C standard and
revisit CFP 1849. See [Cfp-interest 1920] "numerically equal" and "numerically equivalent" David Hough

Jim: Numerically equal should take into account the sign of zero. So

-0 can't be the same as +0. Stronger than comparing equal.

Mike: Level 2 in IEEE distinguishes between -0 and +0. Quantum is at level 3. C should decide which one it is talking about for numerically equal.

Jim: Should we avoid "numerically equal"?

David H: Say equivalent instead of numerically equal?

AI: David H.: Look at each use of "numerically equal" and "equivalent" and see what should be changed in the C standard.

David H: For hypot, the proposed change also takes care of the case with two zero arguments.

For DFP, hypot observes the quantum of both arguments, taking the min of both arguments, which is not the same as fabs.

Mike: Perhaps say hypot(x,+-0) -> hypot(x,+0).

Jim: When this was originally written, we thought there was infinities here.

Jim: hypot is not required to be correctly rounded so you could get different values.

Mike: Perhaps say "hypot(x, +-0) is equal to the absolute value of x when x is a number."?

Jim: Sounds good, though this is written in the same spirit as sqrt.

David H: Or you can say hypot(x, +-0) is correctly rounded when x is a number. Or even say the absolute value of x when x is a number, putting together both points.

AI: David H: Rewrite the conclusion in CFP 1920 so it matches surrounding text and says hypot(x, +-0) is correctly rounded when x is a number along with the CFP 1920 proposed text.

Action item details:

Jim/Fred: Go through all the CFP proposals submitted to ensure they were put in the C standard draft (N2596) correctly.

Not done.

Fred: Submit CFP 1869 to WG14 with the typo fix of adding a space to the second last change.

N2640 2021/01/30 Tydeman, Missing DEC_EVAL_METHOD

Fred: Resend the document relating to updating the example in G.5.1 using max to use the
newer functions as a new proposal.

[Cfp-interest 1896] fmax, fmin Fred J. Tydeman

Jim: Seems better to move the note (first change) to after fmin/fmax or after the f{max/min}imum_num subclauses.

Perhaps remove the word NOTE there since it is a sub-clause itself.

Jim: 7.31.8 should not be needed since we already obsoleted the functions.

Fred: OK.

Jim: F.3#3 seems like a good change.

AI: Fred: Make CFP 1896 into a proposal for WG14 removing the word "NOTE" in change 1, and doing the change questioned in 7.31.8.

Fred: Submit CFP 1870 to WG14.

N2641 2021/01/30 Tydeman, Missing +(x) in table

Fred: Send out the changed text (CFP 1891) to 754.

AI: Fred: Send out the email numbers for handling the CFP 1891 action item

Rajan: Test the change in CFP 1891 with IBM's implementation.
[Cfp-interest 1899] Re: WG14 IEEE 754-C binding meeting minutes 2021/01/13
AI: Fred: When presenting CFP 1891, ensure that the hypot case is said to not apply.

Fred: Submit the change proposed in CFP 1891 to WG14.
N2642 2021/01/30 Tydeman, Quantum exponent of NaN

Jim/Fred: Make CFP 1866 into a draft WG14 proposal with the change "Positive zeros compare
equal to negative zeros."
[Cfp-interest 1901] AI about comparing zeros Jim Thomas
Mike: 754 is good with this. There were uncertainties before.
AI: Jim: Submit the document in CFP 1901 to WG14.

Fred: Change "negative signed value" to "a value with a negative sign" in CFP 1886 and submit it to WG14.
N2643 2021/01/30 Tydeman, Negative

Jim: Make CFP 1881 into a WG14 proposal removing "with a negative sign bit".
[Cfp-interest 1903] AIs about negative values Jim Thomas
AI: Jim: Submit the paper in CFP 1903 to WG14.

Jim: Look at other uses of the phrase "sign bit" in the C standard.
[Cfp-interest 1904] AI about sign bit in C Jim Thomas
AI: David H: Look to see if setpayload{sig} in IEEE 754 says anything about the sign bit.
Fred: If 754 doesn't say anything about that, it could be an errata in 754.
David H: We seldom say anything about the sign of NaNs.

Jim: Submit CFP 1879 to WG14 changing "reports" to "determines" in the footnote.
N2650 2021/01/30 Thomas, C2X proposal - signbit cleanup

Rajan: Look into whether "should" can be used in an ISO standard. (Yes, allowed: See https://www.iso.org/foreword-supplementary-information.html)

Jim: Submit CFP 1883 as a WG14 proposal.
N2651 2021/01/30 Thomas, C2X proposal - fabs and copysign cleanup
[Cfp-interest 1916, 1921] Fwd: (SC22WG14.18893) fabs, copysign, representations and N2651 Jim
Thomas
Jim: The last two changes can go forward while we look at the first two.
WG14: Rajan: Only propose the last two changes (not the bit representation changes).
Ian: Should fabs and copysign be specified the same way as IEEE in C?
Jim: They were sign bit operations originally in IEEE for perf reasons?
David H: Yes, that is correct.
Jim: Is the bit level semantics important anymore?

David H: No, doesn't seem to be.
Ian: For performance.
Jim: We could say the interchange format encodings are values in the type. Different encodings are different values in the type.
Ian: That would imply +0 and -0 are different values.
Jim: Yes, they are now, and NaNs would be as well and non-canonical encodings would also be different values.
Jim: We could add a footnote to C to say it could talk about representation level while 754 does it at the encoding level.
David H: Originally we considered very fast implementations.
Jim: Is there a reason a function like fabs can't be implemented at a bit level.
It doesn't fit the conceptual model, but allowing it to return non-canonical results.
Leave this item until next time.

Jim: Get permission from WG14 to revise TS 18661-5 along the lines of what is proposed for TS
18661-4 in CFP 1856.
[Cfp-interest 1907] AI re permission for TS 18661-5 revision Jim Thomas
Jim: Submitted to WG14.

Fred: Follow up with the WG14 editor about the changes in CFP 1874.
Fred: Still no response from the editor on these.
Carry over

Jim: Combine CFP 1880 and CFP 1885, and update CFP 1885 to address the and/or ambiguity
in the first suggested change and put it in the form of a proposal.
[Cfp-interest 1908] AI about 5.2.4.2.2 cleanup Jim Thomas
AI: Jim: CFP 1908: Change "fk == 0" to "fk = 0".
Fred: Should the (signed or unsigned) be before infinities?
Jim: Didn't want it to apply to NaNs. Could put NaN's first.
AI: Jim: CFP 1908: Make a change to put NaNs before infinities: "not floating point numbers, such as NaNs and (signed or unsigned) infinities."
AI: Jim: CFP 1908: Remove the bit-representation paragraph's second sentence.
AI: Rajan: Review Jim's update to CFP 1908 before submission to WG14.

Other issues:
Errata for IEEE 754-2019
[Cfp-interest 1914] Errata for IEEE 754-2019 Mike Cowlishaw
(http://speleotrove.com/misc/IEEE754-errata-2019.html)
Mike: Will be removing the 9.2.1 item since it is not an errata. It can go into David H's futures document.
AI: Mike: Bring forward the IEEE errata (CFP 1914) after removing the third item to CFP before bringing it to IEEE.

cr_xxx functions
[Cfp-interest 1905] cr_xxx functions Paul Zimmermann
[Cfp-interest 1906] Re: cr_xxx functions Jim Thomas
Jim: We don't have them as reserved names since they are not 754 operations.

Are we good with reserving them?
Fred: See if "cr_" as a prefix is reserved or in the process of being reserved.

Domain errors for quiet and signaling NaNs?
[Cfp-interest 1915] Domain errors for NaNs? Jim Thomas
Jim: No reason to require SNaNs to be a domain error.
Fred: Calling the function would cause the signal so the function wouldn't ever see the SNaN.
So we should not require it to cause domain errors.
Ian: Using SNaNs to represent some other values could have functions do an operation without causing a domain error.

Propagation of NaN sign bit
[Cfp-interest 1911] propagation of NaN sign bit Paul Zimmermann
David H: There were some deficiency issues with their implementation. It is software based, so they should preserve it if it is essentially free.
Ian: Is there a cost to not preserving it?
David H: There was an implementation where the sign had semantic meaning. It is not contrary to IEEE.

Range errors
[Cfp-interest 1841] C math errors Jim Thomas
[Cfp-interest 1842] Re: C math errors Fred J. Tydeman
[Cfp-interest 1843] Re: C math errors Jim Thomas
[Cfp-interest 1873] Range error Fred J. Tydeman
[Cfp-interest 1912] Re: C math errors Jim Thomas
[Cfp-interest 1913] Re: C math errors Fred J. Tydeman
Did not get to this item

Parameterization of interfaces
Did not get to this item