WG14 N2705
Meeting notes

C Floating Point Study Group Teleconference

2021-03-24
8 AM PDT / 11 AM EDT / 3 PM UTC

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

New agenda items:
None.

Carry over action items:
Jim/Fred: Go through all the CFP proposals submitted to ensure they were put in the C standard draft (N2596) correctly. - Done!

Last meeting action items (all done unless specified otherwise below):
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.

New action items:
Fred: Make a proposal for CFP 1927 with the change of the final change being "equal" instead of boolean and check with David H.
Fred: Write up CFP 1930 as a proposal.
Fred: Submit CFP 1938 to WG14.
Fred: Check with the CFP group (and possibly others) to see if the default static initialization gives all zero bits for DFP values.
Fred: Send the IEEE 754 errata note that is currently not reflected in the errata list to the CFP group.
Mike: Check what the zero bits with the bias exponent means for DFP (regarding static initialization). (During the meeting: Mike: 0e-101 is what the result is.)
Fred: Create a WG14 proposal to reserve either cr_ or reserve specific cr_{function name}s as per CFP 1906 and let WG14 decide.

Study group logistics:
Next meeting date: Tuesday, April 13th, 2021.
8 AM PDT / 11 AM EDT / 3 PM UTC
Same teleconference number.

C++ liaison:
Issues? None.

WG14 meeting report:
[Cfp-interest 1947] WG14 202103 results of CFP papers Rajan Bhakta
Fred: For the pow case. It is ambiguous in the 1e0 case. I asked the IEEE group. Should it be the 0 or the -Inf quantum exponent?
pow has a preferred quantum exponent of y * Q(x). For x being an integer (no fractional part), Q(x) is 0. If it is a +Inf, it should be "1.0" if it is a -Inf, it should be "1." The 0*Inf is not defined.
AI: Fred: Send the result of N2642's follow up to WG14.
Deadlines for new proposals for C23: Spring next year.
Deadlines for updated proposals for C23: Fall next year.

C23 integration:
Latest C2X drafts:
http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2596.pdf
http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2573.pdf
http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2478.pdf
Part 1
Part 2
Part 3
Part 4ab
Part 5abcd
IEC 60559:2020 support

Action items from 2021-02-17 meeting:
Carry over: Jim/Fred: Go through all the CFP proposals submitted to ensure they were put in the C standard draft (N2596) correctly.
Jim and Fred have given comments to JeanHyde. No response.
Close action item.

David H.: Look at each use of "numerically equal" and "equivalent" and see what should be changed in the C standard.
[Cfp-interest 1927] action item: suggest rewordings for F9.2 of n2596.pdf David Hough CFP
Jim: For the final change, boolean values is not used in C.
David H: Perhaps just say "equal".
Fred: Both operators return an integer, not a boolean.
David H: Do we mean logically equal?
Fred: isless says it is always equal to x < y (with the text about invalid flag).
David H: So we can just say "equal".
AI: Fred: Make a proposal for CFP 1927 with the change of the final change being "equal" instead of boolean and check with David H.

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.
[Cfp-interest 1926] action item: usage of "when x is a number" in n2596.pdf David Hough CFP
[Cfp-interest 1928] action item: suggest wording for hypot David Hough CFP
[Cfp-interest 1929] Re: action item: suggest wording for hypot Paul Zimmermann
[Cfp-interest 1931] Re: action item: suggest wording for hypot Jim

Thomas
[Cfp-interest 1930] Re: action item: suggest wording for hypot David Hough CFP
Jim: We're putting in a case here that's actually covered and I don't know if we've done this for other functions.
David H: I'm pretty sure it's not.
Jim: What is unusual about hypot is the NaN case is specified otherwise rather than taking the global statement. Ex. The third case in the email. pow is a case like this (pow of zero). Do we put in a special case for NaN there?
David H: The case of disappearing NaNs is a contentious one so I'm for listing cases like this explicitly.
Fred: pow says it.
Jim: I don't object to this change as long as other functions have the corresponding functions just like this one.
David H: There is a higher chance of making a mistake if we tried to do it for other cases like pow and complex pow.
Jim: Complex wouldn't be an issue since that specification is different.
Fred: I have an issue with the second bullet. This is a change to the existing standard. 1 and 3 are already in the standard. 4 is a new one.
Fred: What is hypot(NaN, 0)?
Jim: Can we hold this for another time? We need to write this as a proposal.
AI: Fred: Write up CFP 1930 as a proposal.
Jim: In CFP 1926 there is still an issue.
"Determine" is used in many other places so this change in isolation is good, but not in general.
David H: Lets not rock the boat.

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.
[Cfp-interest 1938] Re: WG14 IEEE 754-C binding meeting minutes 2021/02/17 Fred J. Tydeman
AI: Fred: Submit CFP 1938 to WG14.

Fred: Send out the email numbers for handling the CFP 1891 action item.
[Cfp-interest 1923] Submissions on preferred quantum - came up in current teleconference David Hough CFP
[Cfp-interest 1944] Fwd: (SC22WG14.19047) Preferred quantum exponent for default-initialized Fred J. Tydeman
[Cfp-interest 1945] Re: Fwd: (SC22WG14.19047) Preferred quantum exponent for default-initialized 0 Fred J. Tydeman
Jim: All encodings for DFP have zero bits giving zero values right?
Fred: Yes. Though it is not zero quantum exponent. It is the most negative one (-99999 due to biased exponent). I hope implementations set all bits to zero.
AI: Fred: Check with the CFP group (and possibly others) to see if the default static initialization gives all zero bits for DFP values.
Mike: I believe it is a zero with a zero quantum exponent.

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.
N2670 2021/02/27 Thomas, C23 proposal - zeros compare equal

Jim: Submit the paper in CFP 1903 to WG14.
N2671 2021/02/27 Thomas, C23 proposal - negative values
Fred: Is +0 positive?
Jim: No, it says in the text it has to be > 0.
Fred: We should copy the first suggested change and change negative

to positive and less than 0 to more than zero.
David H: More compactly, "Thus zeros and NaNs are neither positive or negative."
Jim: This has already been submitted.
David H: Lets not gild the lily.
Fred: We may debate this in WG14 as an editorial change, but it's good enough.

David H: Look to see if setpayload{sig} in IEEE 754 says anything about the sign bit.
[Cfp-interest 1925] action item: 754-2019 setpayload and setpayloadsig vs. sign bit David Hough CFP
Mike: For me, the sign bit is part of the payload. For an integer the sign determines if the integer is positive or negative.
Jim: I'm not sure 754 allows you to put the sign on the payload.
David H: That's right, the payload is always non-negative.
Mike: In test cases or converting to character strings, I say SNaN (-3) for example.
Fred: The current setpayload functions say they create a quiet NaN with the given payload and a zero 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.
N2672 2021/02/27 Thomas, C23 proposal - 5.2.4.2.2 cleanup

Mike: Bring forward the IEEE errata (CFP 1914) after removing the third item to CFP before bringing it to IEEE.
[Cfp-interest 1937] Re: Errata for IEEE 754-2019 Fred J. Tydeman
[Cfp-interest 1942] Re: Errata for IEEE 754-2019 Mike Cowlishaw
http://speleotrove.com/misc/IEEE754-errata-2019.html
Mike: I've updated this since last time.
Fred: I sent one an email about pow that is not on the list.
Mike: If you send it again, I can update this.
AI: Fred: Send the IEEE 754 errata note that is currently not reflected in the errata list to the CFP group.
AI: Mike: Check what the zero bits with the bias exponent means for DFP (regarding static initialization).
Mike: 0e-101 is what the result is.

Fred: See if "cr_" as a prefix is reserved or in the process of being reserved.
[Cfp-interest 1936] Re: WG14 IEEE 754-C binding meeting minutes 2021/02/17 Fred J. Tydeman
[Cfp-interest 1939] Re: WG14 IEEE 754-C binding meeting minutes 2021/02/17 Jim Thomas
[Cfp-interest 1940] cr_ Fred J. Tydeman
[Cfp-interest 1941] Re: cr_ Jim Thomas
[Cfp-interest 1905] cr_xxx functions Paul Zimmermann
[Cfp-interest 1906] Re: cr_xxx functions Jim Thomas
Fred: cr_ was not reserved, but I have written a proposal to do it.
Jim: Do we want to do the blanket reservation or just add to the list of function names?
Fred: Blanket seems easier.
Jim: Jens had a proposal that reserved a number of prefixes and suffixes, and that proposal was not accepted. We shouldn't do the same

since WG14 didn't move on it.

Damian: Why didn't Jens proposal pass? It seemed there was a lot of stuff that everyone would agree to with other stuff that wouldn't.

Jim: Reserving the prefix takes more out of the namespace.

Rajan: We could propose both and let WG14 decide.

Fred: I can propose the cr_ reservation and say since the list is continually expanding it makes sense to do it this way.

Jim: I'm reluctant to launch into this. We could get a list of the cr_ functions we would add and propose the two alternatives.

AI: Fred: Create a WG14 proposal to reserve either cr_ or reserve specific cr_{function name}s as per CFP 1906 and let WG14 decide.

Other issues:

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

Follow up of CFP 1841:

Issue 4: With alternate exception handling, we need to clean up what raising a floating-point exception means (signaling underflow or raising the underflow flag). The distinction was ignored before, but now it can't.

Fred: Shouldn't that be "set" a floating-point exception flag rather than "raise".

Jim: Currently the standard uses "raise".

Fred: The C standard says "set" as well.

Fred: Intel vs Motorola had issues about when a trap is taken.

Jim: There's no traps in 754.

David H: The original 754 had traps, but now doesn't.

Jim: For raising a floating-point exception, this does not include the exact underflow case.

Issue 3: What's the difference between loss of accuracy and inexact?

Fred: If a result is the smallest normal, is it underflow?

David H: Depends on whether the rounding mode had a result in between or after. It's too complicated so we don't specify it.

Issue 5: Fred: The change will make the math library slower for some people.

Jim: That's true.

Fred: Must/Shall should stay "must".

Fred: hypot of (double_min, zero) can no longer raise the underflow exception? It should be double_min, but some implementations do square, square root. Or (3*double_min, 2*double_min) will generate underflows.

David H: Fast implementations will have these underflows. What's normally done is scaling.

Jim: There are performance implications to this, but this is an editorial change. It's already there in 7.12.1.

Issue 6: Fred: I always took "mathematical result" meaning infinite precision.

David H: Right. It is the theoretical result.

Mike: Even with correct rounding, you can't tell if the result is out of the range.

Jim: I don't think so.

Mike: I don't think you should change the meaning of "mathematical result".

Jim: Originally it was an alternative, but I nixed it.

Fred: Could say a computed result before rounding.

Mike: You can't say that since it hasn't been computed yet.

Mike: A mathematical result is not a real number. What is here is

not correct. It can be more precise than the correctly rounded result.
Fred: There is also a distinction for real vs complex.

Parameterization of interfaces

Floating-point accuracy in C
[Cfp-interest 1932] Re: Fix the inaccuracy of j0f/y0f/j1f/y1f Paul Zimmermann
[Cfp-interest 1934] Re: Fix the inaccuracy of j0f/y0f/j1f/y1f Vincent Lefevre
[Cfp-interest 1933] Re: Fix the inaccuracy of j0f/y0f/j1f/y1f Mike Cowlishaw