WG14 N2721
Meeting notes

C Floating Point Study Group Teleconference

2021-04-13
8 AM PDT / 11 AM EDT / 3 PM UTC

Attendees: Rajan, Jim, Fred, Ian, Mike, Damian, David O. (left early), David H. (joined late)

New agenda items:
None.

Carry over action items:
None.

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

New action items:
Fred: Submit CFP1963 as a proposal after adding in the positioning of the inserted text.
Fred: Create a proposal for default static initialization regarding the default quantum exponent and send it to the email list.
Fred: Submit CFP 1967 changing it to only offer option 1 and list a rationale of why we didn't add in the other functions to the list.
Jim: Look at the use of the word "normal" in the C standard to ensure the use in range error clarifications is OK.
Jim: Range errors: Issue 1: Instead of "below" in the footnote, use a clause/subclause number.
Jim: Create proposals for range error clarifications.
Rajan: Check with David Keaton on the timing of bug fix/clarification proposals for C23.
Jim: Propose CFP 1952 with the addition of a title for the second table to WG14.
Rajan: Talk to Jim to figure out next steps.
Any: CFP1934: Look into adding in that basic math operations (+, -, *, /) should be correctly rounded.

Study group logistics
Next meeting dates: Wednesday, May 19.
Same time (3pm UTC).
ISO Zoom teleconference

Please notify the group if this time slot does not work.

C++ liaison
None.

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

Fred: End of this year C23 is frozen, so the new feature proposals have to be in by the next meeting and updates by the meeting after. We have more documents than time for the next meeting.
Jim: A lot of our proposals are bug fixes or clarifications, not new features. Does that count?
Rajan/Fred: It was not said.

Action items discussion
Fred: Make a proposal for CFP 1927 with the change of the final change being "equal" instead of boolean and check with David H. (See CFP1968)
Jim: David H. looked at this?
Fred: Yes. He was good with it.
Mike: I think we did the same change in 2005 for IEEE.
Looks good.

Fred: Write up CFP 1930 as a proposal. (See CFP 1963)
Jim: You should say where this is added.
Fred: Does it matter?
Jim: It would help the editor to say “After bullet x”
Fred: OK.
AI: Fred: Submit CFP1963 as a proposal after adding in positioning of the addition.

Fred: Submit CFP 1938 to WG14 (min-max cleanup).
Fred: Waiting for a document number.

Fred: Check with the CFP group (and possibly others) to see if the default static initialization gives all zero bits for DFP values.
Fred: No feedback from any group (IEEE or C).
Rajan: We are adding in a new restriction that wasn’t there before. I’m good with it, but is there something with this may break anyone?
Mike: Having zero with a quantum of zero is not all zero bits.
Jim: If that’s what you wanted, you should provide an initializer.
Fred: We can make it a footnote.
Mike: No need to standardize it.
Fred: So leave the grey part and add a footnote that says we don’t specify what quantum exponent it is, but prefer the least.
Mike: We should just leave it implementation defined what it is.
Jim: We’re converging on a footnote: The quantum exponent is implementation defined. The common implementation of all zero bits give the least quantum exponent.
AI: Fred: Create a proposal for default static initialization regarding the default quantum exponent and send it to the email list.

Fred: Send the IEEE 754 errata note that is currently not reflected in

the errata list to the CFP group.

Mike: Discussion still ongoing. No convergence.

David H: Few people have experience using quantum with these functions. Not much experience to draw on.

Jim: We need to decide on what to do for C. Can we go forward with what Fred has proposed here?

Fred: No, I've changed my mind here. If the computation of the quantum exponent ends up being a NaN then it should be zero.

Jim: The computation of the quantum exponent is real arithmetic, not computational arithmetic. So not sure what NaN's mean there.

Fred: Q(NaN) is a NaN.

Jim: But that is not computational issue. It is not defined.

Mike: I think we agree Q(NaN) is zero but what is Q(Infinity)?

Fred: That's defined by IEEE.

Fred: I will send an email to the CFP group about this.

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.)

Done.

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

Jim: There were more functions than sqrt, like cube root, erf, erfc, {l,t}gamma, etc. so they need to be added to the option 2. Seeing this, I prefer option 1.

Damian: We wanted 1, but we weren't sure that was acceptable so we had option 2.

Jim: It wasn't clear to me last time, but seeing it now, I'm inclined to only do option 1.

Mike: We should add a reason why we chose to do it this way than adding in specific functions. Adding a rationale will help listing we need to add a number of other functions.

AI: Fred: Submit CFP 1967 changing it to only offer option 1 and list a rationale of why we didn't add in the other functions to the list.

Other Issues:

Range errors (See [Cfp-interest 1913] Re: C math errors, wiki.edg.com/pub/CFP/WebHome/range_errors_and_exceptions-20210410.pdf)

Jim: For the performance hit referred to last time by Fred, this is nothing new, it is just a re-specification of what was there already. The implementation just has to check the final result magnitude is not outside the normal range. It is an easy check.

Fred: The footnote 403 change requires something which footnotes can't do.

Jim: That's the point, this is not a new requirement, 7.12.1 says it normatively. Should I say "as implied by 7.12.1" instead?

Rajan: Not good to have imply. Say direct if it is direct.

Fred: I'm good with the words as they are.

Jim: It's C underflow vs. IEEE underflow exceptions.

Issue 6:

Rajan: Any way of making "normal" special so it is not taken as the English normal.

Fred: How about "normalized"?

Mike: That does not apply to Decimal.

Jim: In the abstract sense it does.

Jim: The new change is copied from previous text in 7.12.1.

Damian: The real problem is we need to define "normal".

Jim: Lets go ahead with this as is.

Mike: And separately look at the "normal" issue.

AI: Jim: Look at the use of the word "normal" in the C standard to

ensure the use in range error clarifications is OK.

Issue 7:
Use the second change (minimal: add the word “finite”)

Issue 1:
Mike: Using “below” in a footnote is not good since it may end up being above. Perhaps give a subclause.
AI: Jim: Range errors: Issue 1: Instead of “below” in the footnote, use a clause/subclause number.

Issue 8:
Mike: Couldn’t range error go away and instead just use overflow/underflow.
Jim: It’s used in all the math functions.
Mike: OK, then leave it.

AI: Jim: Create proposals for range error clarifications.
AI: Rajan: Check with David Keaton on the timing of bug fix/clarification proposals for C23.

Inexact exception (See [Cfp-interest 1952] Re: inexact exception Jim Thomas)
Jim: The functions in 60559 are in the operation binding table. The problem is that there are 2 tables, one for exact binding, and a second with mathematical functions but not a complete binding (due to lack of requiring correct rounding or taking into account rounding directions).
Jim: In CFP 1957, changing the F.10 instances to refer to the tables. Paul responded with adding in numbered tables but I don’t see that anywhere in the C standard so I don’t want to invent it.
Mike: We can say “not listed in the previous table”
Jim: Both are previous tables.
Mike: Can use paragraph numbers.
Jim: There are no paragraph numbers. We can put paragraph 20 into it’s own subclause (F.3.2). That was we could refer to the subclause.
Mike: That’s reasonable. The second table should probably have a title.
Jim: Not opposed to that.
Mike: Title it “C functions that do not require correct rounding”.
Jim: We’ll work on it.
Fred: We can ask the editor as well.
Jim: We can propose what I said and also propose a title.
AI: Jim: Propose CFP 1952 with the addition of a title for the second table to WG14.

Parameterization of interfaces
AI: Rajan: Talk to Jim to figure out next steps.

Floating-point accuracy in C (See [Cfp-interest 1934] Re: Fix the inaccuracy of j0f/y0f/j1f/y1f Vincent Lefevre and [Cfp-interest 1933] Re: Fix the inaccuracy of j0f/y0f/j1f/y1f Mike Cowlishaw)
Fred: We do say something in C. It is implementation defined.
Mike: Perhaps add some “should”s into the C standard regarding accuracy.
Damian: For real numbers, too hard for complex.
Fred: Even add in the “should”, no one will change their hardware.
Mike: Software is more interesting anyways.
AI: Any: CFP1934: Look into adding in that basic math operations (+, -, *, /) should be correctly rounded.

Others?

None.