

WG14 N3292

Meeting notes

C Floating Point Study Group Teleconference

2024-06-19

8 AM PDT / 11 AM EDT / 3 PM UTC

Attendees: Rajan, Jim, Jerome, Fred, David, Joshua, Damian

New agenda

items (<https://wiki.edg.com/pub/CFP/WebHome/CFP%20meeting%20agenda-20240619-update2.pdf>):

None.

Previous meeting notes:

See [CFP 3111] (<http://mailman.oakapple.net/pipermail/cfp-interest/2024-May/003125.html>).

Next Meeting(s):

July 17, 2024, 3PM UTC

ISO Zoom teleconference

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

New action items:

Jim: Put Annex G being purely IEEE and other parts into the main standard as a potential work item for CFP on the agenda for next meeting.

Fred: Update issue 1 with the information from today.

Jim: Add C2y Issue 23 onto the agenda for next meeting.

Jim: Submit the proposal for C26 issue 19 (CFP 3118 with base document updated and checked for accurate clause numbers) to WG14.

Rajan: Fix the title in the proposal in CFP 3112. Base document as well from N3219->3220 and ensure the clause numbering is correct. Submit to WG14.

Rajan: Change the base document from N3219->3220 and ensure the clause numbering is correct for CFP 3091. Submit to WG14.

Rajan: Issue 20: Make CFP 3045 into a proposal for WG14.

Jim: Create a proposal for WG14 based on CFP 3074 regarding the HUGE_VAL with suffixes change.

Jim: Make CFP 3080 into a proposal for WG14.

Fred: Move the sub-issues from CFP 3079 about "floating-point number" into Issue 8.

Rajan: Make a proposal for WG14 based on CFP 3091.

Jim: Ensure that "frexp and double-double underflow" is on the agenda for next meeting.

Jim: Ensure that "Midpoint and linear interpolation functions" is on the agenda for next meeting.

Action items to be carried over:

IEEE 754 liaison:

David: Project has been approved by the higher levels, so kickoff meeting should be soon.
Need to choose a chair, which may be Leonard Tsai

C++ liaison:

Midpoint and linear interpolation functions

WG14 meeting:

See [CFP 3119] and follow ups. CFP 3124 has the final summary.

Joshua: I macro definition reworded in <https://www.open-std.org/jtc1/sc22/wg14/www/docs/n3274.pdf>

Damian: In the Python working group there was a big argument between whether it should be i (or I?) or j (or J?), Guido said j/J, and done.

Fred: In Fortran, no imaginary type. Just operating on complex parts.

Jim: Are the semantics complex or pure imaginary.

Damian: Complex in Python. In D, Chapel, and Ada it is pure imaginary.

Joshua: Annex G contains two portions. In N3274, G.4 has complex functions for IEEE arithmetic. G.2 talks about general complex multiplication. That needs to be moved to the main standard.

^Jim: Put Annex G being purely IEEE and other parts into the main standard as a potential work item for CFP on the agenda for next meeting.

Fred: glibc was tested by me and no issue there. They have infinite precision.

C23 integration:

Next WG14 meeting:

September 30, Minneapolis. Mailing deadline is one month before the meeting.

C23 drafts:

C23 working draft n3219 - July 2, 2023 - For CFP review only. Do not distribute.

C2y drafts:

N3220

Carry over action items (Done unless stated otherwise):

Jim/Jerome/Damian: Follow up on C26 issue 1.

Jerome: C26: Issue 1: Revamp to remove the "may" for negative errors.

Jerome: Why do we have a weaker "a pole error *may* occur" instead of "a pole error occurs"?

Jim: Not sure, may be historical.

Rajan: Harder to pass if we make it required.

Fred: We can ask WG14 to vote on it too. Weaker vs stronger.

Jim: We could say "occurs" is more consistent with other domain errors, but incompatible with previous behaviour.

Jerome: For $t\gamma$, since the sign can change based on the direction at zero, you can get only one of domain or pole error. You can't get both. Perhaps say "A domain error or pole error may occur" instead of "*a* pole".

Jim: This occurs in other parts of the standard too.

Jerome: With the exception of the indefinite article.

Jerome: You'd only get a pole error if zero had a sign. Without a sign, you'd get a domain error. People may go to $l\gamma$ and say I want infinities giving domain errors too. Less capable floating point would give less rich error information.

Jerome: The intent is one error, not two.

Jerome: The pole error definition text still needs work. Tried to avoid too many and's/or's in it.

Damian: Get a list of editorial issues in Annex G and send them out for future submission to WG14.

See [CFP 3096, 3097, 3098] Annex F and Annex G - primarily about special cases

Damian: Without imaginary types, some of what I'm saying is redundant. I'll need to remove it.

Damian: The domains for the functions in annex F are given by inequalities. They read nice. Annex G didn't use inequalities and used English words instead with inconsistencies. How can you write $+0 \rightarrow \text{Inf}$? It was hard to write it and read it. Need to write inequalities with $+/-0$? That is the hard part.

David: You can't use inequalities when talking about the sign of zero. It always ends up being a mix of English and math notation.

Damian: Can I use open brackets for example?

David: You can use whatever you want, but it has to fit what others understand it to be. No easy way out of it.

Damian: I will redo it using square brackets and send it to the group.

Jim: Will you change annex F?

Damian: No, not touching annex F.

Jim: The English explanation is more helpful to me than a terminology that is invented and people are not familiar with.

Damian: People are not familiar with (and [?

Jim: For signed zero, people are not familiar with using those brackets for that.

Damian: What if I put in a description at the start?

Joshua: I like Fred's example with the sign: " $+0 \leq x < \text{INF} \text{====} [+0, \text{INF})$ ".

Damian: I agree. Same wavelength.

Joshua: All zeros should always have a sign.

Damian: I'll put it out there. Can go back to English if needed.

Jim: -0 is in that interval. $-0 == +0$.

Damian: Not on the number line.

David: Two equal numbers that are equivalent are different here. It is not strictly mathematical.

Jerome: Explicitly stating that can help and be reasonable.

Jim: In interval arithmetic, do they use the $+0$ and -0 ? The key fact about signed zeros are equal.

Damian: They are equal if you are comparing them. Bit patterns are different.

Fred: I think we need to say that $+0$ means only $+0$, same with -0 and $+/-0$.

Jim: But you need to say how they compare.

Damian: What is the next number after -0? +0!

Jim: No, it is the smallest subnormal for nextup.

Jim: You need to define them using the representation of zero instead of inequalities.

Damian: I tested on two systems -0 nextup gives +0. Towards -Inf for +0 gives -0.

Jim: That's wrong if that's what it is.

Fred: Put the text after issue 4 in c26d into issue 5.

Done.

Fred: Create a C26 issue to clear up 7.12.1#1 to ensure SIGFPE is not interpreted as being allowed for any clause 7 functions.

See: <https://wiki.edg.com/pub/CFP/WebHome/c26e.htm>

Done.

Fred: Made issue 23 for the SIGFPE issue and 24 for 5c.

^Fred: Update issue 1 with the information from today.

^Jim: Add C2y Issue 23 onto the agenda for next meeting.

Jim: C26 Issue 19: Add to the changes for 7.24.1.6 the terms "decimal form" and "hexadecimal form" to paragraph 3's bullets 1 and 2 respectively to make it clear what the changes in paragraph 4 refer to. Also do the changes for issue 19 problems 1 and 2.

See [CFP 3118] and follow ups.

Fred: I did update issue 19 with this, but getting large.

Jim: Basically "defining" decimal form and hexadecimal form and using those words.

Rajan: Weird use of kind of defining a term but not really, then using it but not as a real term (italicized). But OK with it.

Fred: Like it.

^Jim: Submit the proposal for C26 issue 19 (CFP 3118 with base document updated and checked for accurate clause numbers) to WG14.

Jerome: Following on what Rajan said, someone may object to this since it is a sneaky pseudo definition. These are not in the arithmetic.

Jim: They are not identical since wide character versions have a different definition. A central definition would have to account for it.

Action items from previous meeting (Done unless stated otherwise)

Rajan/Jim: Ask WG14 if a footnote (that we will have provided already) is needed for the CFP 3074 change to 7.1.2#7 saying an exception could still occur for an expression evaluation using the replacement.

See [CFP 3112] Macros and floating-point exceptions proposal

^Rajan: Fix the title in the proposal in CFP 3112. Base document as well from N3219->3220 and ensure the clause numbering is correct. Submit to WG14.

Rajan: Make a proposal for WG14 based on CFP 3091.

See [CFP 3109] AI from the 2024/05/22 meeting: Optional NaN

^Rajan: Change the base document from N3219->3220 and ensure the clause numbering is correct for CFP 3091. Submit to WG14.

TS-4 and TS-5 revisions

Still waiting for updated draft of C23. Hanging paragraphs can change subclause numbers.

C2y issues

Issues list

<https://wiki.edg.com/pub/CFP/WebHome/c26e.htm>

Status of

issues: <https://wiki.edg.com/pub/CFP/WebHome/C26%20Issues%20status.pdf>

Fred: I will try to add the WG14 document numbers into the issues list from Jim's status page.

Issue 1 (terms and definitions for math errors): See [CFP 2994,3016,3043,3064,3092 and follow ups].

Issue 5,20 (macro exceptions): See [CFP 3045,3074,3112].

5a: CFP 3112 handles.

^Rajan: Issue 20: Make CFP 3045 into a proposal for WG14.

^Jim: Create a proposal for WG14 based on CFP 3074 regarding the HUGE_VAL with suffixes change.

Fred: Issue 24 for the HUGE_VAL suffixed with the type.

Issue 18 (stdc_want_iec_60559_ext in math.h): See [CFP 3080].

^Jim: Make CFP 3080 into a proposal for WG14.

Issue 19 (strto* and wcsto* wording): See [CFP 3058,3078 and follow ups].

Jim: CFP 3079 raises two other issues. Saying "finite floating-point number" which is a red flag since all floating-point numbers are finite. The problem is it is used in a generic sense instead of the defined term in 7.12.1. Also used in ceil and floor, where it is not intended. Not a model number for those functions.

Damian: What does IEEE say about floating-point number is?

Joshua: Includes infinities but not NaN.

Jim: The difference should be noted somewhere.

Jim: Those two "floating-point number" issues should probably go into Issue 8.

^Fred: Move the sub-issues from CFP 3079 about "floating-point number" into Issue 8.

Issue 21 (parenthesis): See [CFP 3091].

No objections.

^Rajan: Make a proposal for WG14 based on CFP 3091.

Imaginary types:

See [N3206, N3241, N3240, CFP 2979,2997,3018,3019,3032,3053,3055,3083 and follow ups].

N3241 2024/04/14 Gustedt, Introduce complex literals

N3263 2024/05/15 Gustedt, Remove imaginary types v.2

Annex G:

See [CFP 2997,3018,3019,3032,3053,3055,3083 and follow ups].

Others?

Other issues:

frexp and double-double underflow

See [CFP 3100 and follow ups].

^Jim: Ensure that "frexp and double-double underflow" is on the agenda for next meeting.

Floating-point class revisited

See [CFP 3113] a floating point class revisited

Midpoint and linear interpolation functions

<https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2019/p0811r3.html>

See [CFP 3114] Two math functions

^Jim: Ensure that "Midpoint and linear interpolation functions" is on the agenda for next meeting.