

WG14 N2876
Meeting notes

C Floating Point Study Group Teleconference

2021-11-23
8 AM PST / 11 AM EST / 4 PM UTC

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

New agenda items (https://wiki.edg.com/pub/CFP/WebHome/CFP_meeting_agenda_20211123-update2.pdf):
None

Carry-over action items:
None

Last meeting action items (done unless specified otherwise, details below):

Jim: Update the INFINITY macro paper to define INFINITY iff infinities exist in type float, and as an alternative, do what is in the current paper.

Rajan: Bring up CFP's position in WG14's liaison report on not proposing double and long double INFINITY macros despite it being brought up. If WG14 wants it, let us (CFP) know.

Jim: Change N2716's alternative wording proposal to replace the second "=" in the example with "yields" and the "page 450" with the section, sub-clause number.

Jim: Look at the changed text and why some unintentional underlining is present in C23_proposal_-_Normal_and_subnormal_classification-20211008.pdf (Ex. emax).

New action items:

Jim: Write a note to the editor of WG14 to copy the definition of INFINITY or make a pointer to the float.h definition in math.h. Also look at the other macros to see if there are others with the same issue.

Jim: Update CFP2277 linked documents to list November 2021 (not just November) for the WG14 meeting.

Jim: See what can be done to show the latest changes to existing accepted changes in all double-double updates to previously passed CFP papers.

Jim: Put 'model number' into the 5.2.4.2.2 classification paragraphs. Ex. Saying 'model floating point number x'.

Jim: Look at making zero's into a separate paragraph for 5.2.4.2.2 number categories.

Jim: Fix the pronouns (remove?) for the document in the second link in CFP2277 in the motivation section.

Rajan: For CFP2291, update the color to be more readable and add in wctod functions updates and refer to overflow/underflow in 7.12.1 if possible.

Fred: Ask C++ what their issues with *_HAS_SUBNORM are and if they are OK with obsoleting it.

Jim: Add in "(finite or infinite)" to the last proposed change after "number" in CFP2280.

Jim: Consider nextafter for the changes in CFP2280.

Jim: Look to put CFP2280 into the overflow update.

Next Meeting(s):

Same time slot.
December 22nd, 2021, 4PM UTC
ISO Zoom teleconference

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

Upcoming WG14 meetings:

January 31-February 4, 2022, Portland, Oregon, US (Tentative)

Mailing deadline: December 31, 2021

July 11-15, 2022, Strasbourg, France (Tentative)

Mailing deadline: June 10, 2022

WG14 meeting report (CFP 2273):

N2797: Fred to write a new paper before doing anything else. Flush operands, etc.

Jim: We have a problematic definition here. Some implementation specific code may depend on this but not a pure portable one. Any consideration on obsolescence?

Rajan: Yes, some. Not sure if everyone agreed.

Jim: For C++ it seems we are supposed to ask if we can remove it.

Fred: Aaron said it won't make the next liaison meeting.

Jim: Any clarification about new proposals?

Fred: No new proposals, but can follow up existing proposals. My tgamma will not go into C23 so no rush on it.

C++ Liaison:

WG21's SC22 special meeting about C/C++ compatibility (See CFP2242/2243):

Ian: I agree with wanting long double + _Float64 -> long double. You'd lose information otherwise.

Jim: No, it wouldn't be the case. If long double is more precise, you get long double.

Ian: OK, makes sense. Then I'm fine.

No response from David Olsen. Leaving it as is assuming agreement or at least no contest.

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>

New draft N2931 is up, but some issues with table of contents and other garbage characters. There will be a newer draft that fixes this.

Action item resolutions:

Jim: Update the INFINITY macro paper to define INFINITY iff infinities exist in type float, and as an alternative, do what is in the current paper (See N2848)

Jim: The change we had only addresses float.h. This should be able to be handled by a note to the editor to copy the definition or have a pointer in math.h to this definition.

^Jim: Write a note to the editor of WG14 to copy the definition of INIFINITY or make a pointer to the float.h definition in math.h. Also look at the other macros to see if there are others with the same issue.

Fred: We may want to have a new paper to do all the macros with a pointer from math.h to float.h.

Jim: Some were not in C18 so we can just remove them.

Rajan: Bring up CFP's position in WG14's liaison report on not proposing double and long double INFINITY macros despite it being brought up. If WG14 wants it, let us (CFP) know.

No issue. WG14 did not ask for it.

Jim: Change N2716's alternative wording proposal to replace the second "=" in the example with "yields" and the "page 450" with the section, sub-clause number (See N2847)

Will be handled when WG14 looks at the paper.

Jim: Look at the changed text and why some unintentional underlining is present in C23_proposal_-_Normal_and_subnormal_classification-20211008.pdf (Ex. emacs) (See N2842)

Jim: A double double update is present. Update coming.

Other issues

Action items from WG14 meeting –5.2.4.2.2 clarification, double-double support (See CFP 2277, 2279, ... 2289)

^Jim: Update CFP2277 linked documents to list November 2021 (not just November) for the WG14 meeting.

CFP2277: Update to N2806: Link 1:

Jim: The changes are cumulative, these are not only the change to what was already accepted. Need to look at how to just do the last change and not the changes to the changes.

Fred: We could do the before/after with the before being the accepted changes as regular text.

Rajan: Could just highlight the newest changes?

^Jim: See what can be done to show the latest changes to existing accepted changes in all double-double updates to previously passed CFP papers.

Fred: Any way to change numbers to model numbers via italics or something? Confuses me all the time.

Jim: How about just saying "model numbers" instead? Or "Floating-point numbers x in the model"?

Fred: Yes, that would work.

^Jim: Put 'model number' into the 5.2.4.2.2 classification paragraphs. Ex. Saying 'model floating point number x'.

David H: I like 'model floating point number x'

Fred: Do we want to cover double-double normalized numbers?

Jim: I don't know what that means, it's already there.

Fred: The strikeout of "f1 > 0 and all possible..." was the definition of the normalized numbers.

Jim: So 1+dbl_min is normalized?

Fred: No, it is not.

Jim: We're adding clarification, not changing the definition of normalized numbers.

Fred: For my change, see CFP2289.

David H: How much we have to say about the numbers outside the model?

Jim: We have the footnote that covers that. We don't want normative text for double-double since there is no standard for it.

David H: We can only standardize the double-double model numbers and leave the rest to their fate.

Fred: OK.

Jim: I also removed 'extra' because there are implementations that have extra range but less precision.

Fred: Do we need a separate paragraph for zero?

^Jim: Look at making zero's into a separate paragraph for 5.2.4.2.2 number categories.

Fred: That can remove a lot of the text in #4.

Jim: No, the #3 paragraphs are definitions, no requirements so #4 stays as is.

Link 2: Overflow/underflow:

^Jim: Fix the pronouns (remove?) for the document in the second link in CFP2277 in the motivation section.

Fred: One objection in the WG14 meeting was someone not understanding "ordinary accuracy".

Jim: Ordinary accuracy was what replaced "extraordinary roundoff error" which was equally vague.

Fred: The HUGE_VAL text is too strong a statement and the however clause in green doesn't override it.

Jim: I can see your point in one sense. We're really making an exception here. There is a change to HUGE_VAL in another topic which may effect this. I don't know which implementations do what. We need this exception for a very small number of exceptions.

Fred: Overflow doesn't mean over the largest normalized, correct?

Jim: Yes. Too large for representation with full precision.

Fred: GCC's double double have full precision for the largest number larger than the largest normalized.

Jim: Yes, that was Josephs specific request that those cases not overflow. I assume that's what those implementations would do, but I don't even know that.

Fred: I withdraw my comment about model numbers in the black text here. What is there is fine.

Link 3:

Fred: The "may" should be a "shall".

Jim: We have an accepted definition and we're trying to stretch it for a certain class of double-double implementations. They have a contiguous set of values with full precision up to that higher precision. The looseness of the specification makes me think "may" is better. Also the implementations may want to classify the values as something else which is allowed for numbers outside of the model.

Fred: Can be may/should/shall.

Ian: I am good with may.

Rajan: 'Shall' may have objections.

Fred: 'Should' can be used (recommended practice type).

Ian: I don't think double-double should be recommended practice.

Fred: I think if it has full precision it should be a normal number.

Leave it as 'may'.

Link 4: Max exponent macros

Looks good.

Action item from WG14 meeting – address INFINITY macro contradiction

Done above.

Action item from WG14 meeting – revise N2823 (Freestanding + FP) (See CFP 2291)

Fred: Lime-green is unreadable. Needs to change.

Fred: wcstod also needs to be done. Maybe just refer to overflow/underflow as defined in math.h.

Jim: We need to make the overflow definition in math.h to be compatible with the strtod functions. If you can point to the other definition for overflow (7.12.1) which we talked about, it would be ideal. HUGE_VAL only matters for strtod and wcstod.

Rajan: I can update this and send it out as soon as I can.

^Rajan: For CFP2291, update the color to be more readable and add in wcstod functions updates and refer to overflow/underflow in 7.12.1 if possible.

Jim: Don't try to expand this out.

Fred: atof has strtod being the same except for error conditions. We may need to expand it there.

Rajan: 7.22.1#1 says we don't need to worry about errno so no issue for this paper.

Action item from WG14 meeting – issues with N2797 (*_HAS_SUBNORM==0 implies what?)

Fred: Should be expanded to allow flushing operands or results independently.

Rajan: I think keeping the macro with new values (as independent/powers of 2) will be palatable for WG14.

Fred: Changing the behavior at runtime should require the value -1. This is beyond the standard.

Removed as an action item: Fred: Write a proposal for *_HAS_SUBNORM updates that allow independent flushing of operands or results as new values for the macro, and saying that changing the behavior at runtime should require value -1.

Jim: That is not my understanding on where we're going with this. I thought you were going to C++, and obsolesce it without change.

Fred: I wanted to give the liaison group the option of leave as is, obsolesce, remove

immediately, or expand.

Jim: I don't know we're good with anything here beyond that.

^Fred: Ask C++ what their issues with *_HAS_SUBNORM are and if they are OK with obsoleting it.

Unordered (unmixable) types (See CFP 2244, 2250, 2251)

Leave as is.

Since Vincent is OK with leaving it as is, we will not propose anything new for post C23.

HUGE_VAL (See CFP 2245's chain)

Jim: For nextup/nextdown we list HUGE_VAL. It is problematic. nextup/downup should step through representable values.

David H: Yes.

Fred: If there is no infinity, is nextup for the largest representable number, is there an overflow or not?

David H: No, it stays without signaling. It is as if it was an infinity.

Jim: For double double, it is representable values, not model numbers.

Fred: So nextup of 1 is the smallest denorm.

Fred: We could add in "(without overflow)" to the suggested text. Also, does 'number' include infinity?

Jim: Yes. My concern with overflow is I'm not sure if overflow is mentioned in that part of the standard. We talk about range errors.

Fred: We can say without a range error.

Jim: We don't say that either.

Jim: Is the '(finite or infinite)' helpful?

^Jim: Add in "(finite or infinite)" to the last proposed change after "number" in CFP2280.

Rajan: Problem 1b's suggested fix: "llibrary" -> "library"

Fred: Can we just say HUGE_VAL is infinity if supported or the largest positive finite number and remove it from the library?

Jim: That wouldn't work for the double double case. Also that change would change the meaning of the macro from being the overflow result. It is not in float.h.

Fred: How about nextafter? Is that an issue?

Jim: We should do something there as well.

^Jim: Consider nextafter for the changes in CFP2280.

Jim: These are tied to the overflow definitions so could get in there. The nextup/down could be considered editorial.

^Jim: Look to put CFP2280 into the overflow update.

Meaning of "nearest" in case of overflow (See CFP 2246, 2260)

Needs discussion.

Continue in next meeting.

Overflow, normalized numbers, N2805 and N2806 (See CFP 2247, 2259 chains)

Covered already.

printf and rounding recommendation (See CFP 2256 chain)

Jim: Leave as Qol?

Continue in next meeting.

Terminology issues (See CFP 2258 chain)

Jim: Need proposal?

Continue in next meeting.

Missing comma (See CFP 2266 chain)

Typo. Already sent to the editor.

tgamma (See CFP 2274 chain)
Fred: New proposal.

Others?

Jim: Fred, if you want to record things that are new proposals, that would be good.
Fred: OK.