WG14 N3229
Meeting notes

C Floating Point Study Group Teleconference

2024-02-07
8 AM PST / 11 PM EST / 4 PM UTC

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

New agenda items (https://wiki.edg.com/pub/CFP/WebHome/CFP%20meeting%20agenda-20240207-update.pdf):
None.

Previous meeting notes:
See CFP2976 (http://mailman.oakapple.net/pipermail/cfp-interest/2024-January/002990.html).

Next Meeting(s):
March 13, 2024, 3PM UTC
ISO Zoom teleconference
Please notify the group if this time slot does not work.

New action items:
Fred: C26: Issue 5: Are there any <math.h> macros with the same issue? Should words be added to an introduction section in <float.h>?
Fred: C26: Issue 9: Look at original CFP messages to see if 3.10 (Correctly rounded definition) might cover it.
Jim: C26: Issue 4: Draft a paper as per the resolution in the issues list.
Jim: C26: Issue 17: Draft a paper as per the resolution in the issues list.
Jerome: C26: Issue 1: Get definitions of terms relating to the issue for 754 and C and regular math.

Action items to be carried over:
None.

C++ liaison:
None

WG14 meeting report
See [CFP2985 and follow ups, CFP2988] WG14 meeting summary
Rajan: Imaginary type removal was voted on by 11/2/10 (consensus). Reasoning was no implementations.
Next WG14 meeting (virtual) is June 10-14th, 2024.

C23 integration

C23 drafts:

C2X working draft n3149 - July 2, 2023 - For CFP review only. Do not distribute.

Status

See [CFP2968] Fwd: [SC22WG14.23987] DIS 9899 passed

DIS review comments (DIS passed)

See [N3216] DIS 9899 Final Disposition of Comments

Editorial review group will meet when we have a draft. CFP signed up to review.

Carry over action items

None

Action items from previous meeting (done unless stated otherwise)

Jim: Update CFP2973 to add a comma before the "or" in the updated note.

See [CFP2978] Fwd: [SC22WG14.24484] DIS comment about correctly rounded result

Jim: Give list of changes compared to the last revision of the TS parts 4 and 5 for the WG14 meeting.

See [CFP2977] list of changes to TSes

Rajan: Issues with the _t names and concrete struct definitions. May be an issue when bringing it into C2Y or future.

Jim, Fred: Come up with a way of discussing these C26 issues (grouping, or something else) before next CFP meeting.

See [CFP299{0,2}] action item about C26 issues

TS-4 and TS-5 revisions

See [CFP2956, 2965, 2989 and follow ups]

Jim: Bill Ash pointed out some fixes that were needed, all in the ISO boilerplate. Those were done and submitted.

Jim: We started in September in 2009, and our objective was to have a C binding to 754:2008. C was on 754:1985. Now we've updated it to 754:2017! And we've brought in a lot into C23 and almost done the latest revisions of the TS's.

C26 issues

Issues list

See https://wiki.edg.com/pub/CFP/WebHome/C26C.HTM

See [CFP2992, 2994 and follow ups]

#5: Is INFINITY for ALL implementations? It is a large FP constant if not supported in C17 (which could raise an FP exception); C23 only if really supported (so not an exception). Exceptions only raised if required (where stated in C23). SNAN quiet or not? Are these (INFINITY, NAN, SNAN) the only special macros?

Hough: As with other macros, these do not raise exceptions when evaluated. NOTE? Footnote? Global FP statement: Unless stated otherwise, none of the <float.h> macros raise FP exceptions when evaluated.

Jim: Are there any <math.h> macros with this issue? Should words be added to an introduction section in <float.h>?

^Fred: Look into.

#9 Issue: 1e99999 to scanf is not infinity.
Jim: correctly rounded definition (3.10) might cover.
^Fred: Look at original CFP messages.

#3 not a problem

#4 OK with resolution
^Jim: Draft paper

#17 round trip requires round to nearest for both conversions.
^Jim: Draft paper. Can be against draft C23.

#1 specific math functions (*gamma); not a problem, really a proposal to tighten up main body.
^Coonen: Math pole vs math singularity; log(0) is not pole, but singularity. 754 might be "wrong", but C matches 754.
Fred: Different implementations did different things, so C std did NOT require just one specific implementation, eg, pole vs domain error for lgamms(0).
C's functions are math functions + useful special cases [pow(0,0), atan2(0,0)] where math is a domain error.

Imaginary types
See [N3206, CFP2979, CFP2997 and follow ups]
Jim: Can address some of Jens comments by moving the semantics of *, - and + into the main body of the standard. Other parts that do not depend on 60559 could also be moved. _Imaginary types could still remain optional.
Joshua: The main standard doesn't have _Imaginary type defined in the main body of the standard. The entire specification is in Annex G.
Jim: Some references in the main body of the standard.
Jerome: We are the only ones doing anything in this space. LAPACK has a complex versions of a lot of its numeric functions. It's been done without imaginary types, so I've been asked how valuable is it? Maybe we should get complex in solidly first before going to imaginary.
Jim: Getting the parts of the operators into the main body would handle that.

Others?

Other issues