WG14 N3055
Meeting notes

C Floating Point Study Group Teleconference

2022-08-12
8 AM PDT / 10 PM EDT / 3 PM UTC

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

New agenda items (https://wiki.edg.com/pub/CFP/WebHome/CFP_meeting_agenda_20220812-update.pdf):
None. Some updates came in late, but will be discussed as they are hit on existing items.

Next Meeting(s):
Cancelling August 24th meeting as it is not needed anymore due to today's meeting.

September 21, 2022, 3PM UTC
ISO Zoom teleconference
Please notify the group if this time slot does not work.

Carry over action items:
Done unless specified otherwise.
Details below in "Carry-over action items results" section.

Last meeting action items:
Done unless specified otherwise.
Details below in “Action items results” section.

New action items:
Rajan: Send Ian information on joining SCC's SC22 mirror committee.
Everyone: If you are in doubt about your membership, contact Rajan, Jim, and Fred.
Jim: I can update Rajan's comment and send it up for review before I submit it Monday.

C++ liaison:
Nothing new.

C23 integration:
New draft available. N3047.
Revised C23 schedule: https://www.open-std.org/jtc1/sc22/wg14/www/docs/n2984.pdf

Study group membership:
See CFP 2517.
David H.: There is no IEEE floating-point committee.
Jim: Anyone who is not a member or is not clear, we can discuss it next meeting.
^Rajan: Send Ian information on joining SCC's SC22 mirror committee.

^Everyone: If you are in doubt about your membership, contact Rajan, Jim, and Fred.

Mike: You can also join ECMA, which is free. Likely Linux foundation is cheap as well.

Rajan: National body's are probably the best. You get to influence ballots as well as joining this group without issue.

Carry-over action items results:

David H: Get an example for the scaled reduction functions (perhaps by asking Jason or Jim or looking into the IEEE references).

Jim: Good to get into for next meeting.

David H: Get an example for the augmented arithmetic functions (perhaps by asking Jason or Jim or looking into the IEEE references).

Jim: Good to get into for next meeting.

Action items results (from previous meeting):

Jim: Post links to the C++ standard in our wiki.

See [Cfp-interest 2474] C++ draft link

Rajan/Jim/Fred/Damien/Mike: Review the CD draft sections as below once it is out to ensure CFP changes were done correctly.

2: References: Fred
4: Freestanding: Rajan
5.2.4.2.2: BFP model: Jim
5.2.4.2.3: DFP model: Mike
6: Language: Jim
7.3: complex.h: Damien
7.6: Floating point environment: Damien
7.12: math.h: Fred
7.22.6: Formatted I/O: <Open>
7.23.1, 7.30.4.1: Numeric conversion: Rajan
7.26: tgmath: <Open>
7.32.4/5/8/13: Future directions: <Open>

See CFP 2477,2479,2481,2482,2483,2487,2493,2494,2499,2501,2504-16,2518,2519.

Jim: Goal: Does CFP agree with submitting the comment as one from CFP? And any change needed? How are we going to submit the comments?

Jim: Suggestion is to keep the name of the comment submitter, but only list the ones we agree to, and submit it as a single document.

Rajan's comments:

Rajan: I agree it should be there in both places (like Jim), but it needs to be there in the first, not the second.

Jim: Can make it two parts. First part is incorrect application of the first part. The second part having it when it shouldn't is that it was an oversight, but we think it should stay.

Fred: The non-negative part was also an oversight.

Jim: Yes, so two oversights. We believe we need both there.

^Jim: I can update Rajan's comment and send it up for review before I submit it Monday.

Fred's comments:

Jim: We had a proposal post published part 2 which was voted in by WG14.

Fred: I didn't check that.

Jim: That's a chore to check.

Mike: If it's reported elsewhere in the standard, we don't need it here as well.

Jim: I agree.

Fred: Looking at the whole standard, it is covered, but math.h alone it is not.

Jim: I suppose someone could use these symbols for completely different reasons which is allowed.

Vivian: DEC_EVAL_METHOD is only defined if IEC_60559's macro is defined.

Jim: Maybe somewhere we need a more blanket specification pertaining to decimal floating types. Something that says it is only there if and only if decimal types are present.

Mike: That is way beyond editorial.

Jim: Agree. OK to leave that for the future?

Fred: Yes, I'm good with that.

fdim:

Mike: Not editorial.

Jim: Seem to be range error cleanup suggestions. We've been making them bit by bit over years.

Mike: The test is, can the editor see clearly that this is true. But this is obscure. Needs justification to make this change.

Fred: Fine to be future.

__suffix__/__prefix__:

Fred: Did see this for embed, but without underscores.

Maybe section:

Jim: Trying to make the general library more in line with Annex F.

Fred: The errors exist in Annex F. Can be next revision.

Unknown section:

Jim: IEEE and C Annex F have a default treatment of NaNs.

Jim: For the type of the function, seems doable.

Mike: A good ballot comment.

Fred: OK with future revision.

M1:

Fred: This is the editors words.

Rajan: Agreed. This should be changed.

Vincent's comments:

Fred: Seems the words are already there in N3047.

Jim's comments:

Complex and 60559:

Jim: This is not a misapplication of a proposal.

Mike: Say "any arithmetic done is as conforming with IEC 60559".

Jim: There is nothing in complex that conforms to the standard.

Mike: Individual parts of calculations are simple arithmetic. I prefer your wording. We want conforming, not compatible though.

Jim: For future.

Same representation/alignment:

Mike: It's a footnote. So not definitive.

Fred: The paragraph is about signed and unsigned integer types.

Mike: It only applies to that point so no need for a change.

Vivian: I agree. Good as is. No need for a change since it is specifically attached to normative text that is for signed/unsigned integers.

Type-generic:

Vivian: Can have a footnote that says it does not have the required properties.

Increment/decrement:

Jim: We don't talk about preferred exponent anywhere in the operators so it is better to put it in the preferred quantum exponent table.

Mike: Not editorial, but a good change.

Jim: Needs to be submitted somehow.

6.7.1#15:

Jim: Can someone check if this change is correct?

Fred: Yes, I can do it.

Mike: Making it a prime or odd would be better.

David H: I just checked. It is exactly representable in 32-bits.

Vivian: 43200002 is also exactly representable. 432000002 is not representable.

typeof:

Mike: Make editorial.

Shadows:

Rajan: Editorial.

Mike: Doesn't matter since it is a comment in an example.

auto b:

Rajan: The example is correct. It is undefined as a declaration, no need for auto.

Idempotent:

Rajan: Forward reference needed?

Mike: Better to avoid forward references.

Rajan: Cancelled action item: Send Jim a comment about having forward references for 6.7.12.7#3's idempotent and other terms to #6/7/8 as a possibility.

Vivian: Defined idempotent for function call or an evaluation. No forward reference needed.

Rajan: Agreed.

5.2.4.2.2#13:

Mike: You are probably right, but may be too late. Clear enough.

Jim: Put on our future list.

Jim: Will make judgment calls on the rest and send it to the group for review for final sending to WG14 on Monday.

-- Time over, ended the meeting --

Jim: Look again at N2570 to ensure proper integration into C23 was done.

See CFP 2475.

Jim: Check with JeanHeyd on issues found in CFP2455 to ensure it is in before the latest C draft.

See CFP 2477.

Other issues

Quantum exponent for ++ and -- operators

See [Cfp-interest 2466 and replies], CFP 2476.

5.2.4.2.3 and IEC 60559

See CFP 2484 and replies.

Others?

Rajan Bhakta
z/OS XL C/C++ Compiler Technical Architect
ISO C Standards Representative (Canada, USA), PL22.11 Chair
C/C++ Compiler Development