WG14 N2971
Meeting notes

C Floating Point Study Group Teleconference

2022-03-30
8 AM PDT / 10 PM EDT / 3 PM UTC

Attendees: Rajan, Jim, Fred, Ian, Jerome Coonen, Damian, Mike, David H, Vivian Van Loan

New agenda items
(https://wiki.edg.com/pub/CFP/WebHome/CFP_meeting_agenda_20220330-update.pdf):
Jim: Allow floating point in integer constant expressions as a new item.

Next Meeting(s):
April 20th, 2022, 3PM UTC
ISO Zoom teleconference
Please notify the group if this time slot does not work.

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

Last meeting action items:
Done unless specified otherwise. Details below in "**Action item results**" section.

New action items:
David H: Look into why Rajan's email's don't carry the attachment links that work when archived.
Jim: Look into the proposed update to TS Part 4 counted page 13, line 30 to see if "+0" should be "positive zero" and whether we should specify the quantum exponent.
Jim: Proposed update to TS Part 4 scaled functions counted page 16, line 36: Change "of NaN" to "or NaN".
Jim: Look into the proposed update to TS Part 4 counted page 21, line 20: Make the first word plural (specification*s*).

C++ liaison:
See [Cfp-interest 2399, 2400, 2406] C++ and extended floating-point types
Jim: This was something we discussed during our meetings.
Fred: Nothing new.

C23 integration:
No new drafts.

Review draft:
See [Cfp-interest 2319 chain] current C2x draft
Old schedule not to be met. David Keaton to issue a new schedule sometime soon.

Carry-over action items results

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

Fred: No responses that addressed that question.

Action items results (from 2022-03-16 meeting)

Rajan: Put it into the WG14 CFP report the outside request to make 7.12.13 and F<x>'s titles "Floating"->"Fused" that is editorial.

See [Cfp-interest 2396,2397] Re: WG14 IEEE 754-C binding meeting minutes

Fred: I read what Steve posted, and in that it said the base FP unit it was unfused. The optional FE unit was fused.

Fred: Post to CFP the WG14 schedule for C23: N2864. - Done.

N2864 2021/11/01 Keaton, Revised C23 Schedule

Fred: Post the decisions on *HAS_SUBNORM by WG14 minutes to Jim.

See [Cfp-interest 2395] *_HAS_SUBNORM

Jim: Can you do issubnorm with nextup of zero?

Fred: Yes, that would work.

Jim: Obsolesce doesn't require changes for implementations.

Rajan: A lot of discussion on what obsolescent means.

Fred: Look at updating the C26.TXT file to follow what Mike is doing for 754.

Fred: Not done yet.

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

David H: Not done yet.

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

David H: Not done yet.

Other issues

N2823 (freestanding) revision

See [Cfp-interest 2398,2410] N2823 revision 2 (version 8)

Jim: Say a freestanding implementation can have 0.

Rajan: That is not the issue, hosted is not a strict superset of freestanding anymore.

Fred: Any issues with locales still? Not FP issue though.

Rajan: Yes, C locale is the only one required by the standard.

x++ and decimal FP

See [Cfp-interest 2405 chain] x++ and DFP

Fred: C/C++ defines ++ as adding the value 1 to the operand. For DFP, what is the quantum exponent of that 1?

Jim: Some responses from Mike and Vincent, and the gist was have zero as the quantum exponent. In C the words say the expression ++E is equivalent to (E)+=1 where the 1 is in program font meaning an int constant.

Mike: I can't see anyone treating it any other way.

Jim: The conversion from int to decimal gives a preferred quantum exponent of zero. So the rules cover it.

Fred: Joseph brought this up.

Mike: Perhaps say it is adding the integer one instead of using the font.

Jim: Rajan, can you handle this if it comes up again?

Rajan: Yes.

TS 18661-4 update

See [Cfp-interest 2390,2393,2409] post-C23 update for TS 18661-4

Jim: (Re 559 vs 60559) Even going to the website you see different names for the same standard, and sometimes even different numbers.

Mike: The sales site should say which name it should be.

Jim: I'm not sure that is consistent.

Jim: The TS does require Annex F conformance now. The freestanding part needs to be reexamined.

Rajan: Conforming is fine. Strictly conforming should be fine too as long as no static storage is required for the accumulation. Need to think about that.

Jim: Augmented could have tgmath, while the others can't since they take array arguments and we haven't had tgmath for those in the past.

Jim: Some changes for the reduction functions to ensure it synced up with 754. The entire specification is in the description now.

Fred: I thought SNANs are required now in C23?

Jim: No, a macro is defined if you support them.

Fred: Page 13, Line 30, "value +0." Is that an integer 0 full stop, or is it a floating-point zero?

Jim: It is supposed to be a positive zero. Could say "positive zero" instead of "+0".

^AI: Jim: Look into TS part 4 p 13, line 30 to see if "+0" should be "positive zero".

Fred: What is the quantum exponent for the decimal functions?

Jim: We say the preferred exponent for the reduction functions is unspecified.

Mike: I think anyone would use 0 for zeros.

Fred: Sometimes you'd want the most negative exponent.

Mike: That's pretty rare.

Jim: If we want to do that here we could but I'd have to change the "is unspecified".

Jim: For the scaled functions, (page 16, line 36) it doesn't say what is returned in the scaled factor pointer. Nor does 754.

^Mike: Should say "or NaN" not "of NaN".

Jim: Hearing it seems OK to require the scale factor to be zero in those cases.

Jim: (Counted) Page 21, line 19: The exceptions here are dancing around the subtle difference between signaling and raising flags in 754. We could say for alternate exception handling may need to signal inexact.

Mike: Maybe ask the 754 list?

Jim: If you can find a way to do that, can you? C doesn't have alternate exception handling.

David H: The first person to put this in hardware will probably determine how it gets resolved.

Fred: On counted page 21, line 20, should the first word be plural?

Jim: Maybe. I think I took this right out of 754.

Jim: Still a question about enough prior art for augmented arithmetic.

Fred: Isn't the double double implementations the prior art?

David H: It's not just like this. Until the hardware exists, it will not be used for double double.

Jim: Too many special cases to make this efficient for software right?

David H: Yes, this was for new hardware. To do what we wanted would be too expensive in software and be too slow. That was the argument back then.

Others

Jim: Idea from Pavel to allow floating-point arithmetic in integer constant expressions. Seemed out of bounds for this group.

Fred: I had words added to C23 that says that new constant expressions are not new integer constant expressions. Comes up for VLA's vs regular arrays. The words preclude what he wants right now.

Jim: If he wants to pursue this, he needs to talk to a WG14 member to do so.

Fred: I can tell him he needs to join WG14 or have a partner who is to bring forward something like this.

Regards,

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