

## **WG14 N3013**

### **Meeting notes**

## **C Floating Point Study Group Teleconference**

2022-05-25

8 AM PDT / 10 PM EDT / 3 PM UTC

**Attendees:** Rajan, Jim, Fred, Ian, Vivian, David H, Damian

### **New agenda items**

([https://wiki.edg.com/pub/CFP/WebHome/CFP\\_meeting\\_agenda\\_20220525-update.pdf](https://wiki.edg.com/pub/CFP/WebHome/CFP_meeting_agenda_20220525-update.pdf)):

None.

### **Next Meeting(s):**

June 22, 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:**

Everyone: Review the stated corrections for the C26 paper on the wiki.

Jim: Put review of the corrections parts of the C26 paper on the wiki on the agenda.

Jim: Update TS part 4 as per the discussion in the 2022/05/25 meeting (tgmth addition, IEEE conformance required, long double needs to be IEEE).

### **WG14 meeting:**

Fred: No floating point papers presented. Final meeting for C23 papers is the July meeting this year. No later versions of those papers. See N2990 for the list of papers to discuss still in WG14 under section 7. Everything should be integrated into the standard.

Jim: We should get an inventory of what we have.

### **C++ liaison:**

None.

### **C23 integration:**

No new drafts.

### **Carry-over action items results:**

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

See [Cfp-interest 2427] C26 paper on wiki

Jim: Having numbers for the items can help.

Jim: Can we review the corrections in the paper?

Fred: I'm OK with that.

^Action item: Everyone: Review the stated corrections for the C26 paper on the wiki.

^Action item: Jim: Put review of the corrections parts of the C26 paper on the wiki on the agenda.

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

David: Not done yet. The examples are well known, but just need to point them out for the overflow/underflow ones.

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

David: Not done yet.

David H: Look into why Rajan's emails don't carry the attachment links that work when archived.

David: There are newer versions of mailman with other issues but I'll look into that.

Fred: Some of my recent posts seem to be double posted.

No one else seems to have a double post.

David: There is a setting in mailman that can avoid sending you a copy of your own mail.

#### **Action items results (from previous meeting):**

Fred: Create a follow-up paper with the C++ information to re-propose obsolescing \*\_HAS\_SUBNORM macros (See CFP 2422 chain)

Fred: C++ defers to C and they don't care what we do with it.

Fred: Jim objected to changing footnote 28. I say it is not absence, it is indeterminate. It can come in through binary, union tricks, etc.

Jim: Why the first change?

Fred: People don't generally consider both operands and results when considering subnormals. They generally only consider results.

Jim: My concern was this may cause change in implementations. Why are we changing something when we are making it obsolescent.

Fred: Since obsolescent can stay in the standard for a long time, we should be correct.

Jim: If we had correct values for the macros, we wouldn't need to obsolesce it.

Fred: A lot of implementations would be indeterminable which is not very useful.

Jim: For the "and/or" change, I think it is a clarification of the intent.

Fred: The existing footnote 28 ignores operands.

Jim: There is a reason for that. It gives a useful property.

David: The abrupt underflow strictly applies to results. You can ignore your own children, but not other peoples children.

Fred: If all floating point operations never produce a subnormal, how do you get them?

David: You inherit it.

Fred: You can get it via type punning, or reading binary.

Vivian: Isn't that a trap representation resulting in UB?

Fred: No, it is blessed by the standard.

Vivian: If you don't have subnormals, then it would be a trap representation.

Fred: The footnote says there is support for subnormal.

Jim: Applications need to vet their data. The current property allows the application to know no subnormals will be created if they know their input is not subnormals.

Fred: I believe Vincent agreed with my changes to footnote 28. And Jim's changes to "and/or" in footnote 27.

Vivian: My issue is making obsolete yet making changes to it.

Fred: I content footnote 28 has always been wrong. We can get rid of it. Also 27 if you want to and make it obsolete.

David: Doing this right would be a lot more complicated. 2 macros, one for operands, one for results, gradual flush or indeterminate as values.

Fred: It is worse than that since it can be changed at runtime so have to be function calls, not just macros.

Jim: If we offer alternatives, we shouldn't not obsolesce it.

Fred: My view was that WG14 wants us to give a single option.

Jim: Representation as zero means it is zero not a subnormal. Seems to be an oxymoron.

Fred: Representation is IEEE.

Jim: That's what we see, but not necessarily everyone.

Jim: "and/or" should be "and".

Fred: That is fine.

Fred: Should we remove both footnotes and also make it obsolescent?

Jim: Should we offer WG14 the option of keeping the footnotes after saying they are a problem.

David: Why do you need footnotes to explain an obsolete feature? You can make a good case to remove them.

Fred: We should not offer options to WG14.

Jim: We can propose obsoleting it and removing the footnotes.

Fred: OK.

#### **Other Issues:**

TS Part 4 update (<https://wiki.edg.com/pub/CFP/WebHome/cfp4r-20220403.pdf>)

Jim: Should the augmented arithmetic functions be added? Due to the prior art issue. Currently they are in there.

Fred: Yes. The purpose of the TS is to allow people to experiment. The feeling I got from WG14 is that they want more TS's to see if they should add a feature to the standard.

Jim: Should we require annex F conformance? Currently we do.

No issues.

Fred: Only IBM's hex float is different.

Jim: This is different from hardware. This is Annex F.

Fred: Based on my testing, I have yet to find a single implementation that satisfies Annex F.

Jim: Couple reasons possible: Not intending to support it. Or having issues.

Fred: The problems I see are with conversions between floating point to integer.

Jim: Do implementations generally define the macros?

Fred: I don't think anyone does. I assume they define it and see how close they come.

Jim: We need to also look into the spec. Ex. No floating point exceptions if we don't require annex F.

David: The effort to extend IEEE things to non-IEEE systems is vastly difficult. Better to not take it on.

No issues with requiring annex F conformance.

Jim: Disallow these functions for non-IEEE long double formats. The reason being that the functions depend on IEEE arithmetic. Particularly the augmented arithmetic.

No issues.

Fred: You may get pushback on the fact that arrays are before sizes in the existing standard. I happen to like what we have here.

Jim: Are we OK with proceeding the way it is now and change it if WG14 wants?

No issues. OK with the way it is.

Jim: Should we have tgmth versions of the functions? No reason not to.

Rajan: New structs would be needed though for this, but not in the other functions.

Jim: The corresponding real types makes it naturally follow.

No issues with having tgmth macros for the new functions.

^Action item: Jim: Update TS part 4 as per the discussion in the 2022/05/25 meeting (tgmth addition, IEEE conformance required, long double needs to be IEEE).

### **Others?**

None

### **Rajan Bhakta**

z/OS XL C/C++ Compiler Technical Architect

ISO C Standards Representative (Canada, USA), PL22.11 Chair

C/C++ Compiler Development