

## WG14 N2782

### Meeting notes

## C Floating Point Study Group Teleconference

2021-06-23

8 AM PDT / 11 AM EDT / 3 PM UTC

**Attendees:** Rajan, Jim, Fred, Mike, Ian

#### **New agenda items:**

None.

Mike: I did post a 754 errata page, with 52+ viewers, but no comments. Implies we did it right.

#### **Carry over action items:**

None.

#### **Last meeting action items (all done unless specified otherwise below):**

Fred: WG14 N2714 Add "a" before "NaN" in last bullet.

Fred: Paper didn't get discussed since the backlog was so big.

^AI: Rajan: Make this a carry over action item.

Fred: Default static init [CFP 1990]: rewrite as: zero, and if it has DFP type, the quantum exponent is implementation defined; (not sure about "quantum" here) (look at C std to see if bullet list ends in ; not .)

- Do changes, and submit to WG14.

- N2755 2021/06/20 Tydeman, static initialization of DFP zeros

Fred: Similar. Did not get discussed. It takes a long time for a paper to show up to WG14.

Fred: cr\_ prefix [CFP 1967] N2715: Submitted to WG14. Make sure last sentence of paragraph is kept when discuss with WG14.

- N2715 2021/05/09 Tydeman, cr\_ prefix

Fred: In the queue.

Jim: [CFP 1997]: Range error definition as union of overflow and underflow. Change "below" in footnote to "In this subclause".

- Do that change and submit to WG14. Submit to WG14.

- N2745 2021/05/30 Thomas, C23 proposal - range error definition

Jim: [CFP 1997]: Range error definitions of overflow and underflow. Overflow: "ordinary accuracy" not defined, but OK.

- Underflow: exact minimum normal is underflow. Seems wrong. But no better wording.

- Thomas, C23 proposal - overflow and underflow definitions

- N2746 C23 proposal

Jim: See CFP2010. The "or" in there seems confusing, but it can be reconsidered.

Ian: It does seem right to me. Unless there is a better proposal.

Jim: "ordinary accuracy" seems not perfect, but I can't think of anything better.

Rajan: I can see questions on what "ordinary" means.

Jim: Perhaps keep this on the agenda for next time and have people look at it.

^AI: Jim/Fred/David H: Look at N2746 to see if it can be worded better.

Jim: [CFP 1997]: Annex F overflow and underflow

- Rword footnote 403.

- Submit to WG14.

- N2747 2021/05/30 Thomas, C23 proposal - Annex F overflow and underflow

Ian: Seems OK.

Jim: [CFP 1997]: feraiseexcept: traps and alternate exceptions.

- Fix "IECC" and submit to WG14.

- N2748 2021/05/30 Thomas, C23 proposal - effects of fenv exception functions

Jim: Only change was fixing a typo of IECC to IEC.

Jim: Annex F binding: CFP 1998, CFP 2002

- Add 'however' to F.3

- Redo 7.31.8 reference: Reserves cr\_ prefix names for ... N2749 2021/05/30 Thomas, C23 proposal - IEC 60559 binding

Looks good.

Rajan: Post to list results of discussion of next steps w.r.t. shorthand for functions in C standard.

Rajan: Next step is to wait for JeanHeyd to integrate the changes into C23. No response to my note asking if there was anything else CFP needed to do.

Jim, David, Fred: [CFP 1991]: WG14 N2642. "observation" is weird. Breakup in two sentences. Finite result is implementation defined - footnote: not specific in 754, but a possible definition is 0. Rework and resubmit to WG14.

- N2754 2021/06/20 Tydeman, DFP: Quantum exponent of NaN (version 2)

Fred: In the WG14 queue.

#### **New Action Items:**

Jim/Rajan: Look into what we proposed for TS part 4 and the status of it.

Jim: Put the CFP C23 changes summary page for the C++ liaison study group on the agenda for next meeting.

Rajan: Re CFP2024: Ask Paul if there are other values and if it is a real problem.

Fred: Draft a response to the remquo issue brought up in CFP1974.

Fred/Jim: Create a paper re CFP2030 to propose a response.

Rajan: Send N2751 to JeanHeyd.

#### **Study group logistics**

Next meeting date: Wednesday, July 21

Same teleconference number and time.

#### **WG 14 meeting report**

- [Cfp-interest 2029] WG14 meeting results for CFP papers

- [Cfp-interest 2032] WG14 meeting: Other CFP related items Rajan Bhakta

Rajan: Next WG14 meeting is end of August.

Jim: Any clarifications on new proposals?

Rajan: This was supposed to be the last meeting for new proposals, but now is later (October 15th by Fred).

Fred: Mailing for next meeting is July 30th.

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

Inquiry about C++ proposal for "Extended floating-point types and standard names"

Jim: Basic questions about bfloat16 and part 3.

#### **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>

Part 1

Part 2

Part 3

Part 4ab

Part 5abcd

IEC 60559:2020 support

#### **Other issues**

Implementation defined macro values

- [Cfp-interest 2024] values of FE\_TONEAREST... Paul Zimmermann

lan: Making a fixed value would break at least one of the libraries right?

Jim/Fred: Yes.

Mike: We could recommend the common case.

Jim: We could, but not sure there is one.

Rajan: Doesn't solve the portability problem.

Mike: In the long run it does.

Mike: We could ask Paul to get the information about other libraries instead of us saying anything.

Ian: What is the frequency of use of each of the libraries? It would be unfortunate to have us recommend one over the other.

Ian: What does IBM do?

Rajan: WG14 normally cares about source portability, not binary portability. Some exceptions, but generally not. Perhaps ask if there are other values and if there is any actual user issues with this.

^AI: Rajan: Re CFP2024: Ask Paul if there are other values and if it is a real problem.

Jim: I believe the initial reason for this was performance!

remquo unspecified cases

- [Cfp-interest 1974] remquo( ) Fred J. Tydeman

- [Cfp-interest 1975] Re: remquo( ) David Hough CFP

- [Cfp-interest 1979] Re: remquo( ) Paul Zimmermann

- [Cfp-interest 1983] Re: remquo( ) David Hough CFP

Fred: I suggest we add something to Annex F saying the quotient is unspecified.

Jim: It's not the entire quotient, just the low order bits.

Jim: I got the impression that there should be a different definition if  $y == 0$ .

^AI: Fred: Draft a response to the remquo issue brought up in CFP1974.

Fred: Most implementations use the bottom 3 bits of the quotient. Making it INT\_MIN is bad.

Jim: Anyone know of any use of remquo? I think this is a remanent of the old IEEE standard and for use in trig functions.

Mike: We could drop it.

Jim: We could obsolesce it, but not drop it. I don't actually know about the use of it.

Mike: You don't want unspecified or undefined if you don't need them. Anytime you have it, it causes problems for someone.

type\_HAS\_SUBNORM

- [Cfp-interest 2025] FLT\_HAS\_SUBNORM is 0: what is fpclassify(<subnormal>)? Tydeman

- [Cfp-interest 2026] Re: FLT\_HAS\_SUBNORM is 0: what is fpclassify(<subnormal>)? David Hough CFP

- [Cfp-interest 2027] Re: FLT\_HAS\_SUBNORM is 0: what is fpclassify(<subnormal>)? Steve (Numerics) Canon

- [Cfp-interest 2030] Re: FLT\_HAS\_SUBNORM is 0: what is fpclassify(<subnormal>)? Jim Thomas

Fred: At least one person has said operands vs results should cause a split in the macro. Arm has this issue. I believe they always flush to zero regardless of rounding modes instead of the minimum normal value as IEEE says.

Jim: It is not just results, it is results when there is an underflow signal. It is the only time it happens. Ex. copysign onto a subnormal value doesn't end up zero.

Jim: There is confusion on the bit-representation of things. See Cannon's response re making subnormals non-canonical zeros. This would make HAS\_SUBNORM be zero.

Fred: That's what I want. The hardware may do that, but I don't know what the macros do for that implementation.

Jim: If that's what the hardware does, and what it wants the software wants done, that's fine. What do we change? For comparisons, they would all compare equal. Does ARM do that? If they don't do operands but do flush results, then they would treat them as subnormals.

Fred: For ARM, the C macro would have to be -1, not zero. User programs can change the control bits on the fly. They are not protected.

Jim: That's fine. Lots of implementations have that.

Jim: The other model is the 754 abrupt underflow that only affects results and only if there is an underflow signal. That would require saying they support subnormals but requires particular behavior.

Fred: Are exact subnormals underflow? What causes an exact subnormal to underflow? If an underflow does not signal, it is not abrupt.

Jim: An exact subnormal, because it is tiny, has an underflow signal.

Ian: If you had a value that is subnormal, but a bit near the high end of the fraction set, all others being zero. If you divide it by 2 a different bit is set, and the result is exact. I believe all hardware will treat it as an underflow even though it is exact.

Jim: The underflow signal is done for 754.

Ian: Since it is exact, it is not truly an underflow for a human. The hardware knowing that would be very hard.

Jim: The hardware needs to get it right. We're trying to make the C underflow more general than IEEE since it has to deal with other implementations. In C you'll never see a signal without the flag. All you can do is test for the flag. Because there is no alternate exception handling. Abrupt underflow is an alternative exception handling mode.

Fred: If the result is tiny and non-zero, it is underflow. For abrupt, it is flushed (whether exact or not).

Jim: If there is no underflow signal, the result is not signaled (Ex. copysign, nextup)

Fred: nextup of zero?

Jim: nextup does not signal.

Jim: It is only the way the exception is handled that is different. It can handle subnormals so nextup of zero can give a subnormal.

Fred: The purpose of the macro is if a programmer does arithmetic operations, can they get subnormal results. And if they have subnormal operands and do arithmetic on it, is it treated as a value that is non-zero or as a zero. Both have to be supported for a value of the macro to be 1. If either one is true but not the other, it has to be -1.

Fred: Any nextup of a subnormal has to give the minimum normal. nextdown of any subnormal would have to be all bits zero.

Jim: Only if you want canonical results.

Jim: Steve's model is an implementation model. You can't require it.

Rajan: Why not answer this as a clarification request. Say -1 for mixed results and hence the classification macro results for subnormals will be whatever the implementation wants it to be.

Jim: That would mean that IEEE abruptUnderflow would have an indeterminable result.

Fred: Correct.

^AI: Fred/Jim: Create a paper re CFP2030 to propose a response.

Signbit cleanup

- N2751 2021/06/20 Thomas, C23 proposal - signbit cleanup with typo fix

^AI: Rajan: Send N2751 to JeanHeyd.

Others?

None.