**WG14 N1998**
**Meeting notes**

## C Floating Point Study Group Teleconference

2016-01-12
9 AM PST / 12 PM EST

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

**New agenda items**:
Maintenance to the published parts of the TS
Part 1 comments from Fred

**Last meeting action items**:
Ian: Talk to Lawrence Crawl regarding proposing this IEEE-754: 2008 binding to C++ as well. - In process.
No response from Lawrence to an email sent by Ian.
Ian: Update and check the items listed and flagged under Feature_List_Part_1. - Not done.
David: See what Oracle has done for the feature lists. - Done
Not much present. They do have Fred's test suite. No DFP yet.

Rajan: Get Mike in contact with Jens to get access to the C Standard source in the GIT repository. - Done.
Copy Jim on this as well.
Jim: p13: line 22: Add in a statement describing the remaining bullets to show they are not a complete list. - Done.
Jim: p17: line 27: List the actions in place of "below" or say "the remaining actions in this sub-clause". - Done.
All: Consider mixed nested try and delayed try blocks and look for cases not covered by the specification. - Done. Not reflected in the current draft.
All: Focused Review of part 5 once the updated draft comes out. - Delayed to handle comments from Joseph.

**New action items**:
Jim: Part 5: Page 2: Review the change to see if clarification is needed regarding "after" referring to time or source code.
Jim: Part 5: Page 18: Line 45: Reword to make it clearer.
Jim: Part 5: Page 20, 21: Change the '#define n' to '#define NUMBER_OF_ELEMENTS'.
Jim: Part 5: Page 21: Change the "for (i=0;" to "for (int i=0;"
Jim: Get the changebar'd and dated version of previous TS's on the wiki to review.

**Next Meeting**:
February 16th, 2016, 12:00 EST, 9:00 PST
Same teleconference number.

**Discussion**:
WG14 mailing is March 14th, 2016. Meeting is April 11-15th, 2016 at BSI, London, England.

IEEE 754 revision:

Progress being made.
Expecting new policies and procedures.

Arith23:
Nothing new.

Part 5: Various emails, documents (cfp5-20160105.pdf [http://wiki.edg.com/bin/login/CFP/WebHome?origurl=/pub/CFP/WebHome/cfp5-20160105.pdf]):
Copyright dates changed (except page ii as that is an image).
*Page 2 change: Review this to see if clarification is needed regarding "after" referring to time or source code.
Saying "after the signal handler completes." will make it unambiguous.
Does this apply to all the standard pragmas?
Seems to be safer to have it apply to everything since there would be no implementation assumptions.
Page 5: Good.
Page 6 change 1: Good.
Page 6 change 2: Good.
Page 9 change 1 (comment 5): Good.
Page 9 change 2 (comment 6): Good.
IEEE does not say anything regarding to this.
For alternate exception handling, it is specified.
For optimization allowance it is not.
Page 10: Good.
Page 15 change 1 (line 7): Good.
Fits better with existing C language since it does not exclude complex.
Page 15 change 2 (comment 8): Good.
Page 15 change 3 (comment 7, 9):
Any requirement to make the values distinct?
No. The use is still for the list of exceptions. The values are a form to allow testing for existence.
Fred: On page 18 line 45 they seem to need to be distinct.
Ex. The IBM implementation has sub-exceptions that set the parent exception flag bit as well.
*Jim: That is an assumption. Maybe the words need to be changed to be clearer.
Page 18: Good (barring above).
Fred's comment/email regarding jumps (line 21): C standard defines jumps (6.8.6) without longjmp and raise.
Page 19 handles raise(SIGFPE) but not any other raise.
Note this also applies to any other non-returning function calls like quick_exit, etc.
The C Standard definition is for the grammar, not the general english term being used here.
Page 19 change 1: Good.
Page 19 change 2 (comment 13): Good.
Fred: Do any of our TS parts tie together exceptions and SIGFPE?
No. Not specified by us. The C standard already has words about how it can interact and result in undefined behaviour.
We don't want to get into signalling here if we can avoid it.
What does 'scope' refer to here? The FENV_EXCEPT pragma given in the line before.
Page 20: Good.
Suggestion: Change the '#define n' to '#define NUMBER_OF_ELEMENTS'
Page 21: Good.
Suggestion: Change the "for (i=0;" to "for (int i=0;"

Page 22, 23 (comments 8, 11, 14): Good.
If the sub-exception did not set the parent exception, this would break this code.
The nested immediate 'try's have undefined behaviour (what is y's value?) due to the jump out of the inner 'try'.
Fred: Perhaps on page 18 we should list nested try blocks as part of the jumps list as well.
This would be like listing everything else too. Why just this case?
Another possible example would be ALL_EXCEPT with a particular exception.
Perhaps have examples with other exceptions?
Recent versions of LAPACK do exploit IEEE exception handling by using isnan.

Comment 9: Probably is common in implementation of libraries.
Note that sub-exceptions are a optional part of an optional TS so should not be that large a burden.

Are we ready for the focused review after the changes listed above?
Yes.

Maintenance to the published parts of the TS:
Some changes are present in the editors draft (but not the published one).
Jim to get the changebar'd dated version on the wiki to review.
After approval, need to create a DR for the changes.
Editorial can be all in a single DR per TS.
Other changes can be individual DR's.

Parts 1 and 2 implementations in practice:
Should we bring up individual features from these parts to the C standard for inclusion?
Part 1 has a lot of new functions that narrow the results of the arguments. Not a lot of implementation seen.
Jim: Intel may have implemented this.
Fred: No floating point exception flags for DFP implemented in Intel.
Jim: HP did.
Having the entire TS parts is a high bar. Not suitable for an Annex either since it is modifications to the standard.
Ian: Having the syntax for the new floating point literals should be brought in.
Rajan: The TS's should have implementation somewhere (as long as the aggregate of implementations cover the entire TS) to help get WG14 approval to bring it into the standard. Individual proposals of parts of the TS's for standardization is fine, but we should not do it as a group.
Fred: The DEC alpha chip does static rounding.
May not be like ours.
Make this a more significant item to discuss in the next meeting.

Part 1 comments from Fred (email on 2016/01/05):
Seem valid comments.
Review these items next meeting (follow up in email until Fred leaves).

Regards,

Rajan Bhakta
z/OS XL C/C++ Compiler Technical Architect
ISO C Standards Representative for Canada
C Compiler Development