

WG 14 N1920

2015/03/17, 12:00 PM EDT / 9:00 AM PDT:

Attendees: Rajan, Jim, Fred, Vincent, David, Ian

New agenda items:

None.

Sticky action items:

None.

Last meeting action items:

David: Part 5: Write up a proposal for what the FENV_REPRODUCIBLE pragma means. - Done (by Jim)

David: Part 5: Write up a proposal for alternate exception handling. - Done (by Jim)

Jim: Part 5: p4: Line 27: The STDC pragmas should be FENV_ prefixed. - Done

Jim: Part 5: p5: Line 20: $x * y$ should be $x + y$. - Done

Jim: Part 5: p5: Line 33: $x / y = x * (1 / z) \rightarrow x / y = x * (1 / y)$. - Done

Jim: Part 5: p6: Add a note saying that the optimizations do not have to be consistent. - Done

Jim: Part 5: Some dash's have become hyphens throughout the document. - Done

Jim: Part 5: p6: Add a note saying 754 says synthesis where we say contract. - Done

Jim: Part 5: p7: Need example for contract operation conversion. Ex. $\text{float } a = b * c$ (where b, c can be double's, and allow a single rounding). - Done

New action items:

Jim: cfp5-diff-20150211-20150309.pdf: p4: Line 28: Check if 'indeterminate' is the right term.

Jim: cfp5-diff-20150211-20150309.pdf: p8: State what the default state is for #pragma STDC FENV_REPRODUCIBLE and what happens when it is OFF.

Jim: cfp5-diff-20150211-20150309.pdf: p9: Line 5: program code -> something else such as 'code sequence' or 'code segment' or 'Following are requirements for code in the whole program or part of a program'.

Jim: cfp5-diff-20150211-20150309.pdf: p9: Line 10: Note that we need to make sure functions with two or more arguments (since the order of evaluation of them is not fixed) is handled.

Jim: cfp5-diff-20150211-20150309.pdf: p9: Line 32: The other fused multiply functions need to be listed here.

Jim: cfp5-diff-20150211-20150309.pdf: p9: Line 36: Remove "propagation, "

Jim: cfp5-diff-20150211-20150309.pdf: p9: Line 37: Look into tightening the underflow and inexact part.

Jim: cfp5-diff-20150211-20150309.pdf: p10: Line 12: Change the statement to make sure that style 'a' applies to the entire statement.

Jim: cfp5-20150317.pdf: p10: Line 8: unspecified -> indeterminate

Jim: cfp5-20150317.pdf: p10: Line 16: unspecified -> indeterminate

Jim: cfp5-20150317.pdf: p10: Line 16: Remove duplicated text.

Jim: Part 5: Reproducibility: Add in don't use alternate exception handling.

Jim: Make the updates to part 5 and poll the group to see what to do for the C meeting.

Next Meeting:

April 7th, 2015, 12:00 EST, 9:00 PDT

Same teleconference number.

Discussion:

Part 1: Published.

Part 2: Published.

Was published before we had a chance to review ISO edits.
The document will have to be republished.
Waiting for draft from ISO to make fixes.

Part 3: USA has balloted already, Canada will ballot before the end of the month, international is still open.

Part 4: USA has balloted already, Canada will ballot before the end of the month, international is still open.

Part 5:

What should we do for Part 5 with regards to the WG14 meeting?
Consensus to not present it for a vote since it is not anywhere near ready for that.
Better to send it even though we know it needs a lot of work or wait until we get a more solid document.
We are taking the preliminary comments and reacting to them.
May have ISO work item timing issues.
We could show the parts of part 5 that are relatively solid to show progress.

cfp5-diff-20150211-20150309.pdf:

p4: Line 28: indeterminate -> unspecified. Since indeterminate can have traps.
*ToDo: Jim: Part 5: p4: Line 28: Check if 'indeterminate' is the right term.
Does disallow * mean no optimization is allowed even if there is no change in result?
No. The "as-if" law applies.
Is it clear that you cannot reassociate integer associations (due to overflow)?
We are only dealing with the floating point. The regular standard should deal with this.
An operation that produces the same value but with different exceptions: Should this be listed?
The "as if" rule applies here too.
Should this be covered under the value changing optimizations?
No, since it is not changing the value.
This means that value changing optimizations can change the exceptions.
Note that different compilers may have exceptions emitted in different order. This means you may not have reproducible behaviour even with reproducible behaviour turned on.

p8: Nothing said about the pragma state being off.

p8: Nothing said about the default state.

*ToDo: Jim: Part 5: p8: State what the default state is for #pragma STDC FENV_REPRODUCIBLE and what happens when it is OFF.

*ToDo: Jim: Part 5: p9: Line 5: program code -> something else such as 'code sequence' or 'code segment' or 'Following are requirements for code in the whole program or part of a program'.

*ToDo: Jim: Part 5: p9: Line 10: Note that we need to make sure functions with two or more arguments (since the order of evaluation of them is not fixed) is handled.

*ToDo: Jim: Part 5: p9: Line 32: The other fused multiply functions need to be listed here.

p9: Issue 1: Basically means if not depending on the exception, then it is OK to use it.

p9: line 36: Why is quiet NaN listed?

Jim: You do have to return qNaN's but it is not necessarily the same one.

Fred: Payloads or sign bits cover everything.

*ToDo: Jim: Part 5: p9: Line 36: Remove "propagation, "

p9: line 37: Why underflow?

Jim: underflow is optional (before or after) in IEEE. Not sure about inexact.

David: Not sure why this is in the IEEE standard.

*ToDo: Jim: Part 5: p9: Line 37: Look into tightening the underflow and inexact part.

p10: Line 10: Issue 2: We are differing from IEEE here.

*ToDo: Jim: Part 5: p10: Line 12: Change the statement to make sure that style 'a' applies to the entire statement.

What is the issue with remquo?

Implementation defined how many bits are returned as part of the quotient.

Is this what we are looking for for this section?

Consensus is yes.

cfp5-20150317.pdf:

Date is wrong on alternate pages.

p9: line 29: May happen for operations some times and not others (not always for the same operation).

p10: line 8: unspecified -> indeterminate. Due to unsequenced non-atomic updates possibly resulting in trap representations.

*ToDo: Jim: Part 5: p10: Line 8: unspecified -> indeterminate

*Issue: Part 5: p10: Line 4: What is termination of the compound statement? If the compound statement is a function?

*ToDo: Jim: Part 5: p10: Line 16: unspecified -> indeterminate

*ToDo: Jim: Part 5: p10: Line 16: Remove duplicated text.

p10: line 25: Issue: *<Missed this part due to phone issues>

Suggestion: Put the name of the action before the description.

Following the C standard layout hence the way it is now.

Could do liaison report or posting it, or posting it but not making it a paper to review in the meeting.

Can explain the 4 areas we are talking about without mentioning the details, but it may not register with people.

*ToDo: Jim: Part 5: Reproducibility: Add in don't use alternate exception handling.

*ToDo: Jim: Make the updates to part 5 and poll the group to see what to do for the C meeting.

Rajan Bhakta

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

ISO C Standards Representative for Canada

C Compiler Development

Contact: rbhakta@us.ibm.com, Rajan Bhakta/Houston/IBM