Proposal for C2X  
WG14 N2559

Title: Update to IEC 60559 2020  
Author, affiliation: C FP group  
Date: 2020-09-01  
Proposal category: Editorial  
Reference: N2478, N2531, N2532

The suggested changes below update draft C2X (N2478) to the latest version of the floating-point standard: IEC 60559:2020 (IEEE 754-2019). For background on the update see N 2531.

The suggested changes include:

- Updates to the references to the floating-point standard. (Change #1 - 3)
- An update to the introduction to Annex F. (Change #3)
- Corrected/sharpened qualifications. (Change #4, 5, 14)
- Changes to move bindings from the remarks following the binding table in F.3 #1 into the table itself. (Change #6 -13)
- Clarification about how C functions are represented in the binding tables in F.3. (Change #5, 14)
- New IEC 60559 operations (corresponding to existing C functions) added in the binding table in F.3 #23. (Change #15)

N2532 includes changes to accommodate the new min-max functions in IEC 60559:2020.

**Suggested changes:**

1. Change 2 #6:


   to:


2. Change footnote 23:

   23) IEC 60559:1989 specifies quiet and signaling NaNs. For implementations that do not support IEC 60559:1989, the terms quiet NaN and signaling NaN are intended to apply to values with similar behavior
3. Change F.1 #1-2:


to:


4. Change footnote 380:

380) Since NaNs created by IEC 60559 arithmetic operations are always quiet, quiet NaNs (along with infinities) are sufficient for closure of the arithmetic.

5. In F.3 #1, change the first sentence:

[1] C operators, functions, and function-like macros provide the operations required specified by IEC 60559 as shown in the following table. In the table, C functions are represented by the function name without a type suffix.
6. In the operation binding table in F.3 #1, add:

<table>
<thead>
<tr>
<th>getPayload</th>
<th>getpayload</th>
<th>F.10.13.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>setPayload</td>
<td>setpayload</td>
<td>F.10.13.2</td>
</tr>
<tr>
<td>setPayloadSignaling</td>
<td>setpayloadsig</td>
<td>F.10.13.3</td>
</tr>
</tbody>
</table>

7. In the operation binding table in F.3 #1, add:

<table>
<thead>
<tr>
<th>quantize</th>
<th>quantize</th>
<th>7.12.15.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>sameQuantum</td>
<td>samequantum</td>
<td>7.12.15.2</td>
</tr>
<tr>
<td>quantum</td>
<td>quantum</td>
<td>7.12.15.3</td>
</tr>
</tbody>
</table>

8. In the operation binding table in F.3 #1, add:

<table>
<thead>
<tr>
<th>encodeDecimal</th>
<th>encodedec</th>
<th>7.12.16.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>decodeDecimal</td>
<td>decodedec</td>
<td>7.12.16.2</td>
</tr>
<tr>
<td>encodeBinary</td>
<td>encodebin</td>
<td>7.12.16.3</td>
</tr>
<tr>
<td>decodeBinary</td>
<td>decodebin</td>
<td>7.12.16.4</td>
</tr>
</tbody>
</table>

9. Delete F.3 #9:

[9] The C `getpayload`, `setpayload`, and `setpayloadsig` (F.10.13) functions provide program access to NaN payloads, defined in IEC 60559.

10. Delete F.3 #14:

[14] The `quantized`N functions (7.12.15.1) provide the quantize operation defined in IEC 60559 for decimal floating-point arithmetic.

11. Delete F.3 #18:

[18] The `samequantum`N functions (7.12.15.2) provide the sameQuantum operation defined in IEC 60559 for decimal floating-point arithmetic.

12. Change F.3 #20:

[20] The `quantum`N (7.12.15.3) and `llquantexpd`N (7.12.15.4) functions (7.12.15.4) compute the quantum and the (quantum) exponent q defined in IEC 60559 for decimal numbers viewed as having integer significands.
13. Delete F.3 #21-22:

[21] The `encodedc`N(7.12.16.1) and `decodedc`N(7.12.16.2) functions provide the encodeDecimal and decodeDecimal operations defined in IEC 60559 for decimal floating-point arithmetic.

[22] The `encodedb`N(7.12.16.3) and `decodedb`N(7.12.16.4) functions provide the encodeBinary and decodeBinary operations defined in IEC 60559 for decimal floating-point arithmetic.

14. In F.3 #23, change the first sentence:

[23] The C functions in the following table provide mathematical operations recommended by IEC 60559 and similar operations. The C functions are represented by the function name without a type suffix.

15. In the binding table in F.3 #23, change the three rows:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>tanPi</code></td>
<td>tanpi</td>
<td>7.12.4.14, F.10.1.14</td>
</tr>
<tr>
<td><code>asinPi</code></td>
<td>asinpi</td>
<td>7.12.4.9, F.10.1.9</td>
</tr>
<tr>
<td><code>acosPi</code></td>
<td>acospi</td>
<td>7.12.4.8, F.10.1.8</td>
</tr>
</tbody>
</table>