Proposal for C2Y
WG14 N3364 (revision 3 of N3288)
Title: Give consistent wording for SNAN initialization
Author, affiliation: C FP group
Date: 2024-09-30
Proposal category: Editorial
Reference: Base document: N3220

Changes from N3288 (1st version):

- As per WG14 preference (to include constexpr), the "translation time" wording is selected instead.

Proposal:

SNAN used at translation time has different wording in two places: 5.2.4.2.2#22 mentions SNAN for translation time initializer, while H.3#6 mentions static or thread storage duration. There should be consistent wording in both places. In the following, "translation time" phrase is used to maintain consistency. In addition, binding of the "optional unary + or - operator" phrase may be confusing so it is moved to make it more clear that you can have +, -, or no preceding operator for a signaling NaN macro.

Regarding proposed change alternative 1:
  If the standard says "evaluated at translation time" is it clear that the reference is to semantics and not a possible optimization?

E.g. For

```
#pragma STDC FENV_ACCESS OFF
auto double x = SNAN;
```

The evaluation of SNAN is semantically runtime, but could be done at translation time as an optimization because of the pragma. However, the (semantically runtime) evaluation of SNAN results in a quiet NaN regardless of when the implementation evaluates it.

Proposed changes:

Alternative 1:

Change H.3#6 (in N3220) from:

If an optional unary + or - operator followed by a signaling NaN macro is used for initializing an object of the same type that has static or thread storage duration, the object is initialized with a signaling NaN value.

To:

If a signaling NaN macro, optionally preceded by the unary + or − operator, is used as an initializer that is evaluated at translation time to initialize an object of the same type, the object is initialized with a signaling NaN value.

Change 5.2.5.3.2#28 (in N3220) from:

If an optional unary + or − operator followed by a signaling NaN macro is used as an initializer that is evaluated at translation time, the object is initialized with a signaling NaN value.

To:

If a signaling NaN macro, optionally preceded by the unary + or − operator, is used as an initializer that is evaluated at translation time to initialize an object of the same type, the object is initialized with a signaling NaN value.

Alternative 2 (original):

Change H.3#6 (in N3220) from:

If an optional unary + or − operator followed by a signaling NaN macro is used for initializing an object of the same type that has static or thread storage duration, the object is initialized with a signaling NaN value.

To:

If a signaling NaN macro, optionally preceded by the unary + or − operator, is used for initializing an object of the same type that has static or thread storage duration, the object is initialized with a signaling NaN value.

Change 5.2.5.3.2#28 (in N3220) from:

If an optional unary + or − operator followed by a signaling NaN macro is used as an initializer that is evaluated at translation time, the object is initialized with a signaling NaN value.

To:

If a signaling NaN macro, optionally preceded by the unary + or − operator, is used for initializing an object of the same type that has static or thread storage duration, the object is initialized with a signaling NaN value.