

Proposal for C2X WG14 N2356

Title: Update for payload functions
Author, affiliation: C FP group
Date: 2019-01-09
Proposal category: Standards coordination
Target audience: Debugging

The following is an update to subclause F.10.13 which was added to C2X by TS 18661-1 plus DR 12. This update addresses an issue raised in WG 14 about the use of “unsigned” and it brings the payload specification into line with IEEE 754. The changes for IEEE 754 are editorial, except the return value is -1 instead of “unspecified” when the `getpayload` input is not a NaN.

F.10.13 Payload functions

IEC 60559 defines the *payload* to be information contained in a NaN. The payload is intended for implementation-defined diagnostic information about the NaN, such as where or how the NaN was created. The implementation interprets the payload as a non-negative integer suitable for use with the functions in this subclause, which get and set payloads of a quiet or signaling NaN as information encoded in part of the NaN significand. ~~The payload can be interpreted as an unsigned integer. The payload is intended to represent implementation-defined diagnostic information about the NaN. The functions in this clause enable getting and setting payloads. The implementation may restrict which payloads are admissible for the user to set.~~

F.10.13.1 The `getpayload` functions

Synopsis

```
[1] #define __STDC_WANT_IEC_60559_BFP_EXT__
    #include <math.h>
    double getpayload(const double *x);
    float getpayloadf(const float *x);
    long double getpayloadl(const long double *x);
```

Description

[2] The `getpayload` functions extract the payload of a quiet or signaling NaN input and return it as a positive-signed floating-point integer ~~integer value of the payload of a NaN input and return the integer as a floating-point value. The sign of the returned integer is positive.~~ If `*x` is not a NaN, the return result is ~~-1~~ `unspecified`. These functions raise no floating-point exceptions, even if `*x` is a signaling NaN.

Returns

[3] The functions return ~~a floating-point representation of the integer value of~~ the payload of the NaN input ~~as a positive-signed floating-point integer.~~

F.10.13.2 The `setpayload` functions

Synopsis

```
[1] #define __STDC_WANT_IEC_60559_BFP_EXT__
#include <math.h>
int setpayload(double *res, double pl);
int setpayloadf(float *res, float pl);
int setpayloadl(long double *res, long double pl);
```

Description

[2] The `setpayload` functions create a quiet NaN with the payload specified by `pl` and a zero sign bit and store that NaN in the object pointed to by `*res`. If `pl` is not a floating-point integer representing an [admissible](#)-[valid](#) payload, `*res` is set to [+0](#)~~positive-zero~~.

Returns

[3] If the functions stored the specified NaN, the functions return a zero value, otherwise a non-zero value (and `*res` is set to [+0](#)~~zero~~).

F.10.13.3 The `setpayloadsig` functions

Synopsis

```
[1] #define __STDC_WANT_IEC_60559_BFP_EXT__
#include <math.h>
int setpayloadsig(double *res, double pl);
int setpayloadsigf(float *res, float pl);
int setpayloadsigl(long double *res, long double pl);
```

Description

[2] The `setpayloadsig` functions create a signaling NaN with the payload specified by `pl` and a zero sign bit and store that NaN in the object pointed to by `*res`. If `pl` is not a floating-point integer representing an [admissible](#)-[valid](#) payload, `*res` is set to [+0](#)~~positive-zero~~.

Returns

[3] If the functions stored the specified NaN, the functions return a zero value, otherwise a non-zero value (and `*res` is set to [+0](#)~~zero~~).