WG14 N1096

Introduction

This document presents a further set of TR 18037 defects, and will be presented and discussed during the WG14 meeting in Lillehammer (April 2005). The numbering follows the issue numbering 1 through 18 from document WG14 N1087 and hence starts with Defect 19.

Defect 19:

Problem: Sec 7.18a.2 introduces a set of typedefs, and describes a convention for the return type of bits'*fx*': either int_*fx*_t, or uint_*fx*_t.

Sec 7.18a.6.5 lists the 12 functions for bits'*fx*', all of which make use of the form 'int_*fx*_t'. According to 7.18a.2 the last six should be of the form 'uint_*fx*_t'.

Solution: change the last 6 function prototypes in the Synopsis of 7.18a.6.5 to:

```
uint_uhr_t bitsuhr(unsigned short fract f);
uint_ur_t bitsur(unsigned fract f);
uint_ulr_t bitsulr(unsigned long fract f);
uint_uhk_t bitsuhk(unsigned short accum f);
uint_uk_t bitsuk(unsigned accum f);
uint_ulk_t bitsulk(unsigned long accum f);
```

Defect 20:

Problem: the text on the countls function in 4.1.7.3 and 7.18a.6.4 reads:

> The integer return value of the above functions is defined as follows:
>
> - if the value of the fixed-point argument f is non-zero, the return value is the largest integer k for which the expression f<<k does not overflow;
> - if the value of the fixed-point argument is zero, an integer value is returned that is at least as large as N-1, where N is the total number of (nonpadding) bits of the fixed-point type of the argument.
>
> Note: if the value of the fixed-point argument is zero, the recommended return value is exactly N-1.

In the 'argument is zero' case, for a signed fixed-point type, the notion 'nonpadding bits' includes the sign bit (see 6.2.6.3); this implies that the N for signed types is one larger that the N for the corresponding unsigned types; this is wrong (it suggests that shifting into the sign bit does not generate an overflow). In stead of '(nonpadding) bits', the notion 'value bits' should be used.

Solution: in 4.1.7.3 and 7.18a.6.4, replace in the 2nd bullet '(nonpadding)' by 'value'.

Defect 21:
Problem: the text on the countls function in 4.1.7.3 and 7.18a.6.4 reads:

> The integer return value of the above functions is defined as follows:
>
> - if the value of the fixed-point argument f is non-zero, the return value is the largest integer k for which the expression f<<k does not overflow;
> - if the value of the fixed-point argument is zero, an integer value is returned that is at least as large as N-1, where N is the total number of (nonpadding) bits of the fixed-point type of the argument.
>
> Note: if the value of the fixed-point argument is zero, the recommended return value is exactly N-1.

From the definition it is clear that for instance
countlsur(UFRACT_EPSILON) == (UFRACT_FBIT - 1)
and
countlsk (ACCUM_EPSILON) == (ACCUM_IBIT + ACCUM_FBIT - 1)
If the text ' (nonpadding) bits' is replaced by ' value bits' (see Defect 20), then the text requires that
countlsr(0.0r) >= (N - 1)
where the latter value equals countls(FRACT_EPSILON).
This seems counterintuitive; one would expect the value of countlsr(0.0r) to be one less than countls(FRACT_EPSILON).

Solution: change in 4.1.7.3 and 7.18a.6.4 the text of the 2nd bullet and the Note as follows:

> - if the value of the fixed-point argument is zero, an integer value is returned that is at least as large as N, where N is the total number of value bits of the fixed-point type of the argument.
>
> Note: if the value of the fixed-point argument is zero, the recommended return value is exactly N.

Defect 22:
Problem: the bitwise integer to fixed-point functions (in 7.18a.6.6) do not use the int_*fx*_t and uint_*fx*_t integer types; the text in 4.1.7.5 is already correct.

Solution:

- change the Synopsis section of 7.18a.6.6 to read:

```
#include <stdfix.h>
short fract hrbits(int_hr_t n);
fract rbits(int_r_t n);
long fract lrbits(int_lr_t n);
short accum hkbits(int_hk_t n);
accum kbits(int_k_t n);
long accum lkbits(int_lk_t n);
unsigned short fract uhrbits(uint_uhr_t n);
unsigned fract urbits(uint_ur_t n);
unsigned long fract ulrbits(uint_ulr_t n);
unsigned short accum uhkbits(uint_uhk_t n);
unsigned accum ukbits(uint_uk_t n);
unsigned long accum ulkbits(uint_ulk_t n);
```

- remove from the of 7.18a.2 the words ' as return types'
- change in 7.18a.2 the first sentence after the list to read:

The integer types **int_*fx*_t** and **uint_*fx*_t** are the return types of the corresponding **bits*fx*** functions and are chosen so that the return value can hold all the necessary bits; the ***fx*bits** functions use these integer types as types for their parameters.