NAME

fma, **fmaf**, **fmal** **-** fused multiply-add

LIBRARY

Math Library (libm, -lm)

SYNOPSIS

#include <math.h>

double
fma(*double x*, *double y*, *double z*);

float
fmaf(*float x*, *float y*, *float z*);

long double
fmal(*long double x*, *long double y*, *long double z*);

DESCRIPTION

The **fma**(), **fmaf**(), and **fmal**() functions return (x * y) + z, computed with only one rounding error. Using the ordinary multiplication and addition operators, by contrast, results in two roundings: one for the intermediate product and one for the final result.

For instance, the expression 1.2e100 * 2.0e208 - 1.4e308 produces infinity due to overflow in the intermediate product, whereas fma(1.2e100, 2.0e208, -1.4e308) returns approximately 1.0e308.

The fused multiply-add operation is often used to improve the accuracy of calculations such as dot products. It may also be used to improve performance on machines that implement it natively. The macros FP_FAST_FMA, FP_FAST_FMAF and FP_FAST_FMAL may be defined in <*math.h*> to indicate that **fma**(), **fmaf**(), and **fmal**() (respectively) have comparable or faster speed than a multiply operation followed by an add operation.

IMPLEMENTATION NOTES

In general, these routines will behave as one would expect if x * y + z were computed with unbounded precision and range, then rounded to the precision of the return type. However, on some platforms, if *z* is NaN, these functions may not raise an exception even when the computation of x * y would have otherwise generated an invalid exception.

SEE ALSO

fenv(3), math(3)

STANDARDS

The **fma**(), **fmaf**(), and **fmal**() functions conform to ISO/IEC 9899:1999 ("ISO C99"). A fused multiply-add operation with virtually identical characteristics appears in IEEE draft standard 754R.

HISTORY

The **fma**() and **fmaf**() routines first appeared in FreeBSD 5.4, and **fmal**() appeared in FreeBSD 6.0.