This paper summarizes the changes that are expected to make it into the next version of the C standard (“C23”) from a number of TS’s relating to the binding of 2008 versions of IEC 60559 and IEEE 754 into C. In particular:

TS 18661-1: Binary floating point arithmetic (required by IEC 60559)
TS 18661-2: Decimal floating point arithmetic (required by IEC 60559, supersedes ISO/IEC TR 24732)
TS 18661-3: Interchange and extended types (optional by IEC 60559)
TS 18661-4a: Supplementary mathematical functions (optional by IEC 60559, reduction functions under 4b not added to C)

A summary of each TS is given below (part numbers correspond to the TS name after the “-“).

Part 1: Binary floating point

Macros conditionally added to give integer type widths.

Macros and functions conditionally added to query and set floating point environment flags and modes.

Macros and functions conditionally added (ex. fromfpx, roundeven, fmaxmag, llogb, nextup, fadd, ffma, totalorder, canonicalize, setpayload, strfromd including tgmath versions).

Add constant rounding modes: #pragma STDC FENV_ROUND direction - Some standard functions are affected by this (Ex. cos, exp, log, scalbn, cbrt, lgamma, rint, fadd, wcstod, wprintf).

Add macros for signaling NaNs.

Add macros for comparison of floating point values (Ex. iscanonical, issignaling, iszero).

Part 2: Decimal floating point

Distinct types (from float, double and long double) conditionally added for decimal floating point types.

Macros conditionally added to provide information about decimal floating point values (Ex. Min, max values, DEC_EVAL_METHOD).

Macros and functions conditionally added to provide parallel decimal floating point environment flags, functions and modes to binary floating point (Ex. fe_dec_setround, DEC_INFINITY, cosd, expd, fabsd, lround, nextafterd, strtod64). Functions conditionally added to do operations between different decimal floating point widths (Ex. dMadddN, dMmuldN).

Functions conditionally added to get decimal floating point type specific information (Ex. samequantumd, llquantexpd).

Functions conditionally added to convert between different decimal floating point encodings (Ex. encodededcd, decodebind).

Format specifiers added to the printf/scanf family of functions to handle decimal floating point types.

Part 3: Interchange (N) and extended (Nx) types - conditionally normative annex

Binary and Decimal floating point information macros generalized to extended and

interchange types (Ex. FLTN_MAX, DECNX_TRUE_MIN).

Binary and Decimal floating point functions and macros generalized to extended and interchange types (Ex. coshfN, ceilfNx, sinhdNx, dMadddNx, strtofN, FP_FAST_FMADDFN, SNANFN, and tgmath versions).

Decimal floating point specific functions generalized to extended and interchange types (Ex. dMencbindN, strfromencdecdN).

Binary complex and imaginary types generalized to extended and interchange types (Ex. _FloatN _Imaginary, _FloatNx _Complex)

Binary complex floating point functions generalized to extended and interchange types (Ex. cexpfN, crealfNx).

Evaluation method macro values updated to include extended and interchange types (DEC_EVAL_METHOD N for _DecimalN, FLT_EVAL_METHOD N+1 for _FloatNx).

Encoding and decoding functions to allow conversions between non-arithmetic interchange types (Ex. decodefN, dMecndecdN).

Part 4a: Supplementary math functions

New functions conditionally added for binary and decimal, extended and interchange floating point types (Ex. pown, acospifN, exp2m1ldN, compoundndNx).