UCVTF (scalar, fixed-point)

Unsigned fixed-point Convert to Floating-point (scalar). This instruction converts the unsigned value in the 32-bit or 64-bit general-purpose source register to a floating-point value using the rounding mode that is specified by the *FPCR*, and writes the result to the SIMD&FP destination register.

A floating-point exception can be generated by this instruction. Depending on the settings in *FPCR*, the exception results in either a flag being set in *FPSR*, or a synchronous exception being generated. For more information, see *Floating-point exception traps*.

Depending on the settings in the *CPACR_EL1*, *CPTR_EL2*, and *CPTR_EL3* registers, and the Security state and Exception level in which the instruction is executed, an attempt to execute the instruction might be trapped.

31	30	29	28	27	26	25	24	23 22	21	20	19	18 17 16	15 14 13 12 11 10	9 8 7 6 5	4 3 2 1 0
sf	0	0	1	1	1	1	0	ftype	0	0	0	0 1 1	scale	Rn	Rd

rmode opcode

32-bit to half-precision (sf == 0 && ftype == 11)
(FEAT_FP16)

        UCVTF <Hd>, <Wn>, #<fbits>

32-bit to single-precision (sf == 0 && ftype == 00)

        UCVTF <Sd>, <Wn>, #<fbits>

32-bit to double-precision (sf == 0 && ftype == 01)

        UCVTF <Dd>, <Wn>, #<fbits>

64-bit to half-precision (sf == 1 && ftype == 11)
(FEAT_FP16)

        UCVTF <Hd>, <Xn>, #<fbits>

64-bit to single-precision (sf == 1 && ftype == 00)

        UCVTF <Sd>, <Xn>, #<fbits>

64-bit to double-precision (sf == 1 && ftype == 01)

        UCVTF <Dd>, <Xn>, #<fbits>

    if ftype == '10' || (ftype == '11' && !IsFeatureImplemented(FEAT_FP16))

```
integer d = UInt(Rd);
integer n = UInt(Rn);

constant integer intsize = 32 << UInt(sf);
constant integer decode_fltsize = 8 << UInt(ftype EOR '10');
FPRounding rounding;

if sf == '0' && scale<5> == '0' then UNDEFINED;
integer fracbits = 64 - UInt(scale);

rounding = FPRoundingMode(FPCR[]);
```

Assembler Symbols

<Dd>   Is the 64-bit name of the SIMD&FP destination register,
          encoded in the "Rd" field.

<Hd>   Is the 16-bit name of the SIMD&FP destination register,
          encoded in the "Rd" field.

<Sd>   Is the 32-bit name of the SIMD&FP destination register,
          encoded in the "Rd" field.

<Xn>   Is the 64-bit name of the general-purpose source register,
          encoded in the "Rn" field.

<Wn>   Is the 32-bit name of the general-purpose source register,
          encoded in the "Rn" field.

<fbits>   For the 32-bit to double-precision, 32-bit to half-precision
          and 32-bit to single-precision variant: is the number of bits
          after the binary point in the fixed-point source, in the range
          1 to 32, encoded as 64 minus "scale".

          For the 64-bit to double-precision, 64-bit to half-precision
          and 64-bit to single-precision variant: is the number of bits
          after the binary point in the fixed-point source, in the range
          1 to 64, encoded as 64 minus "scale".

Operation

```
CheckFPEnabled64();

FPCRType fpcr = FPCR[];
constant boolean merge = IsMerging(fpcr);
constant integer fltsize = if merge then 128 else decode_fltsize;
bits(fltsize) fltval;
bits(intsize) intval;

intval = X[n, intsize];
fltval = if merge then V[d, fltsize] else Zeros(fltsize);
Elem[fltval, 0, decode_fltsize] = FixedToFP(intval, fracbits, TRUE, fpcr,
V[d, fltsize] = fltval;
```