FCVT

Floating-point Convert precision (scalar). This instruction converts the floating-point value in the SIMD&FP source register to the precision for the destination register data type using the rounding mode that is determined by the *FPCR* and writes the result to the SIMD&FP destination register.

Depending on the settings in the *CPACR_EL1*, *CPTR_EL2*, and *CPTR_EL3* registers, and the current Security state and Exception level, 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
0	0	0	1	1	1	1	0	ftype		1	0	0	0	1	opc		1	0	0	0	0	Rn					Rd				

Half-precision to single-precision (ftype == 11 && opc == 00)

```
FCVT <Sd>, <Hn>
```

Half-precision to double-precision (ftype == 11 && opc == 01)

```
FCVT <Dd>, <Hn>
```

Single-precision to half-precision (ftype == 00 && opc == 11)

```
FCVT <Hd>, <Sn>
```

Single-precision to double-precision (ftype == 00 && opc == 01)

```
FCVT <Dd>, <Sn>
```

Double-precision to half-precision (ftype == 01 && opc == 11)

```
FCVT <Hd>, <Dn>
```

Double-precision to single-precision (ftype == 01 && opc == 00)

```
FCVT <Sd>, <Dn>
```

```
if ftype == opc || ftype == '10' || opc == '10' then UNDEFINED;

integer d = UInt(Rd);
integer n = UInt(Rn);
constant integer srcsize = 8 << UInt(ftype EOR '10');
constant integer dstsize = 8 << UInt(opc EOR '10');
```

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.

<Sn> Is the 32-bit name of the SIMD&FP source register, encoded in the "Rn" field.

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

<Hn> Is the 16-bit name of the SIMD&FP source register, encoded in the "Rn" field.

<Dn> Is the 64-bit name of the SIMD&FP source register, encoded in the "Rn" field.

Operation

```
CheckFPEnabled64();

bits(srcsize) operand = V[n, srcsize];
FPCRType fpcr = FPCR[];
boolean merge = IsMerging(fpcr);
bits(128) result = if merge then V[d, 128] else Zeros(128);

Elem[result, 0, dstsize] = FPConvert(operand, fpcr, dstsize);

V[d, 128] = result;
```

Internal version only: isa v33.64, AdvSIMD v29.12, pseudocode no_diffs_2023_09_RC2, sve v2023-06_rel ; Build timestamp: 2023-09-18T17:56

Copyright Â© 2010-2023 Arm Limited or its affiliates. All rights reserved. This document is Non-Confidential.