FCVTL

Multi-vector floating-point convert from half-precision to deinterleaved single-precision

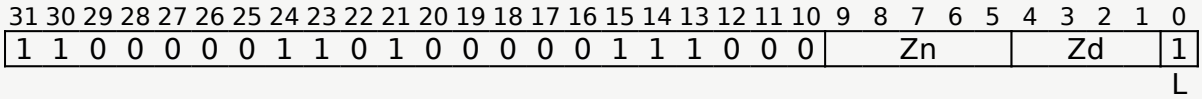
Convert to single-precision from half-precision, each element of the source vector, and place the deinterleaved results in the double-width destination elements of the destination vectors.

This instruction follows SME2 floating-point numerical behaviors corresponding to instructions that place their results in one or more SVE Z vectors.

This instruction is unpredicated.

ID_AA64SMFR0_EL1.F16F16 indicates whether this instruction is implemented.

SME2
(FEAT_SME_F16F16)



FCVTL { <Zd1>.S-<Zd2>.S }, <Zn>.H

```
if !HaveSME2() || !IsFeatureImplemented(FEAT_SME_F16F16) then UNDEFINED
integer n = UInt(Zn);
integer d = UInt(Zd:'0');
```

Assembler Symbols

<Zd1>	Is the name of the first destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 2.
<Zd2>	Is the name of the second destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 2 plus 1.
<Zn>	Is the name of the source scalable vector register, encoded in the "Zn" field.

Operation

```
CheckStreamingSVEEnabled();
constant integer VL = CurrentVL;
constant integer pairs = VL DIV 32;
bits(VL) operand = Z[n, VL];
bits(VL) result0;
bits(VL) result1;

for p = 0 to pairs-1
```

```
    bits(16) element1 = Elem[operand, 2*p+0, 16];
    bits(16) element2 = Elem[operand, 2*p+1, 16];
    bits(32) res1 = FPConvertSVE(element1, FPCR[], 32);
    bits(32) res2 = FPConvertSVE(element2, FPCR[], 32);
    Elem[result0, p, 32] = res1;
    Elem[result1, p, 32] = res2;

Z[d+0, VL] = result0;
Z[d+1, VL] = result1;
```

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