FRINTN

Multi-vector floating-point round to integral value, to nearest with ties to even

Round to the nearest integral floating-point value, with ties rounding to an even value, each element of the two or four source vectors, and place the results in the corresponding elements of the two or four 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.

It has encodings from 2 classes: Two registers and Four registers

Two registers
(FEAT_SME2)

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
1 1 0 0 0 0 0 1	1	0	1 0 1	0 0	0	1 1 1 0 0 0	Zn	0	Zd	0
	size<1>	size<0>								

FRINTN { <Zd1>.S-<Zd2>.S }, { <Zn1>.S-<Zn2>.S }

```
if !HaveSME2() then UNDEFINED;
integer n = UInt(Zn:'0');
integer d = UInt(Zd:'0');
constant integer nreg = 2;
boolean exact = FALSE;
FPRounding rounding = FPRounding_TIEEVEN;
```

Four registers
(FEAT_SME2)

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
1 1 0 0 0 0 0 1	1	0	1 1 1	0 0	0	1 1 1 0 0 0	Zn	0 0	Zd	0 0
	size<1>	size<0>								

FRINTN { <Zd1>.S-<Zd4>.S }, { <Zn1>.S-<Zn4>.S }

```
if !HaveSME2() then UNDEFINED;
integer n = UInt(Zn:'00');
integer d = UInt(Zd:'00');
constant integer nreg = 4;
boolean exact = FALSE;
FPRounding rounding = FPRounding_TIEEVEN;
```

Assembler Symbols

<Zd1>	For the two registers variant: is the name of the first destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 2.
	For the four registers variant: is the name of the first destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 4.
<Zd4>	Is the name of the fourth destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 4 plus 3.
<Zd2>	Is the name of the second destination scalable vector register of a multi-vector sequence, encoded as "Zd" times 2 plus 1.
<Zn1>	For the two registers variant: is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zn" times 2.
	For the four registers variant: is the name of the first scalable vector register of a multi-vector sequence, encoded as "Zn" times 4.
<Zn4>	Is the name of the fourth scalable vector register of a multi-vector sequence, encoded as "Zn" times 4 plus 3.
<Zn2>	Is the name of the second scalable vector register of a multi-vector sequence, encoded as "Zn" times 2 plus 1.

Operation

```
CheckStreamingSVEEnabled();
constant integer VL = CurrentVL;
constant integer elements = VL DIV 32;
array [0..3] of bits(VL) results;

for r = 0 to nreg-1
    bits(VL) operand = Z[n+r, VL];
    for e = 0 to elements-1
        bits(32) element = Elem[operand, e, 32];
        Elem[results[r], e, 32] = FPRoundInt(element, FPCR[], rounding,

for r = 0 to nreg-1
    Z[d+r, VL] = results[r];
```

Base Instructions   SIMD&FP Instructions   SVE Instructions   SME Instructions   Index by Encoding   Sh Pseu